Numerical Methods for Retrieval and Adaptation in Nagao's EBMT model

Kun He
Graduate School of Information, Production and Systems, Waseda University
Kitakyushu, Japan
hekun@asagi.waseda.jp

Tianjing Zhao
Graduate School of Information, Production and Systems, Waseda University
Kitakyushu, Japan
bilibilimisaka@akane.waseda.jp

Yves Lepage
Graduate School of Information, Production and Systems, Waseda University
Kitakyushu, Japan
yves.lepage@waseda.jp

***Abstract*—We build an example-based machine translation system. It is an instance of case-based reasoning for machine translation. We introduce numerical methods instead of symbolic methods in two steps: retrieval and adaptation. For retrieval, we test three different approaches to define similarity between sentences. For adaptation, we use neural networks to solve analogies between sentences across languages. Oracle experiments allow to identify the best retrieval technique and to estimate the possibilities of such an approach. The system could place itself between a statistical and a neural machine translation systems on a task with not so large data.**

***Index Terms*—Example-based machine translation, Numerical methods**

I. Introduction

Example-based machine translation (EBMT) is one of the approaches to machine translation (MT). It is a data-oriented approach. As such, it uses a bilingual parallel corpus as a knowledge base. One of its streams (historical EBMT) is characterised as translation by analogy. It relies on the way language is supposed to be processed in the human brain in the case of primary foreign language learning: no deep linguistic analysis would be needed. EBMT by analogy is lazy learning technique. Its advantage is that the training time is negligible, whereas it is hours for statistical MT (SMT) or even days in neural MT (NMT). NMT requires enormous amounts of data, SMT less, EBMT even less. We choose a scenario where we can honestly compare these three approaches with enough data for all three approaches.

II. Related work

A. Case-based reasoning

Case-based reasoning (CBR) [1] consists in solving a new problem by adapting the solution of an old problem. This requires a case base of old problems with their solutions.

Collins and Way [3] show that EBMT is equivalent to the implementation of CBR in the field of machine translation. To translate a new sentence, similar sentences are looked for and retrieved (retrieve step in [1]) altogether with their translations (reuse in [1]). These translations are adapted to reflect the content of the sentence to translate (revise in [1]). These adapted sentences are checked for validity in the target language before being remembered (retain in [1]) and proposed as translations of the sentence to be translated.

An equivalent view at the relationship between EBMT and CBR [18] identifies three crucial steps: retrieval, adaptation and validation. We follow this view in Sect. III.

The advantage of EBMT is that it dispenses with seeing translation as a rather complex problem for the solution of which features have to be determined and extracted in all generality. It avoids the need for listing up all necessary features or rules for a given language, if translations can be obtained by merely modifying previous translation examples.

B. The development of EBMT

Example-based Machine Translation (EBMT) was initially proposed by Nagao [22]. The claim is that when people translate a simple sentence they do not perform any deep-level grammatical analysis. Instead, they first divide the source sentence into fragments, then translate these fragments into the target language, and finally merge the fragments into one sentence.

In [17] a "purest ever" EBMT system is described. It also uses proportional analogy to capture corresponding but possibly different structures across the source and the target languages without using any grammar rule. The same approach has been applied to the more limited problems of transcribing proper names [12] or unknown words [4].

The European Association for Machine Translation (EAMT) held international workshops on Example-Based Machine Translation three times. These three workshops witnessed the progress in the study of EBMT.

III. Methods and Techniques

In this paper, we examine the introduction of numerical methods in the crucial steps for implementing EBMT as a CBR system. For retrieval, we examine the influence of different criteria in the selection of similar sentences. For adaptation, we use a numerical technique which relies on neural networks. For validation, we conduct an oracle experiment to evaluate the possibilities of our approach.

A. Retrieval

In Nagao's original proposal [22], a sentence is translated by relying on another sentence that differs from it by only one word. As a consequence, in any implementation of this model, the first step is to retrieve sentences which differ the least from the sentence to translate. It means retrieving sentences which are as close as possible to the sentence to translate.

To achieve this, a criterion to compare sentences and estimate their differences so as to be able to rank them by increasing difference is required. By ranking sentences according to differences, it will be possible to select the top N similar ones.

We examine three different criteria to retrieve similar sentences. Dice coefficient, Levenshtein's edit distance and sentence vector representations.

1) Dice coefficient: There exist similarity measures to compare sets. The Dice coefficient is a variation of the Jaccard index [9]. It compares two sets by counting the number of elements in their intersection. To normalize this score, Jaccard index takes the cardinality of the union of the two sets, while Dice coefficient takes the sum of the cardinality of the two sets. As the members in the intersection will be counted twice a factor of two is introduced.

Applied to our problem of computing the similarity between two sentences, we make a set out of a sentence s by considering the set of all its words. We note it by $\overline{s}$. By doing so, we forget about repetition of words and about their order in the sentence. We note the cardinality of a set S by $\mid S \mid$. With these notations, the Dice coefficient between two sentences s_1 and s_2 is defined in Eq. (1).

$$D(s_1, s_2) = \frac{2 \times \mid \overline{s_1} \cap \overline{s_2} \mid}{\mid \overline{s_1} \mid + \mid \overline{s_2} \mid} \tag{1}$$

Jaccard index and Dice coefficient are similarities: the higher, the more similar the two sentences. As said above, they are normalized, hence their values range from 0 to 1. They take a value of 0 when the intersection is the empty set. They take a value of 1 when the two sets are equal. It should be noticed that a similarity of 1 does not necessarily imply that the two sentences are equal.

As an example of computation, consider the previous two French sentences s_1 = *je veux remercier tout le monde.* and s_2 = *je connais tout le monde ici.* Their associated sets are:

$$\begin{aligned} \overline{s_1} &= \{je, le, monde, remercier, tout, veux, .\} \\ \overline{s_1} &= \{connais, ici, je, le, monde, tout, .\} \\ \overline{s_1} \cap \overline{s_2} &= \{je, le, monde, tout, .\} \end{aligned} \tag{2}$$

The cardinalities are:

$$\mid \overline{s_1} \mid = 7,\ \mid \overline{s_1} \mid = 7,\ \mid \overline{s_1} \cap \overline{s_2} \mid = 5. \tag{3}$$

Hence, $D(s_1, s_2) = (2 \times 4)/(7 + 7) = 8/14 \approx 0.57$.

2) Edit distance: The Dice coefficient forgets about the repetitions of words and their order in the sentence. Edit distances can be used to measure how close or far two sequences of words are, while taking into account repetitions of words and their order. They directly measure by how many words two sentences differ. In particular, the Levenshtein edit distance between two sequences of words assigns a score of 1 if only two words are substitute one for the other in the sentences, which is the ideal case in Nagao's model. In all generality, the score of an edit distance is defined as the smallest number of edit operations needed in order to transform a sentence into another. The Levenshtein distance considers three edit operations:

- insertion: a word is inserted somewhere in the sentence. For instance, *Je connais tout le monde.* → *je connais tout le monde ici.* involves the insertion of the word *ici*.
- deletion: a word is deleted somewhere in the sentence. This is the opposite of insertion. For instance, *Je connais tout le monde ici.* → *je connais tout le monde.* involves the deletion of the word *ici*.
- substitution: a word is substituted for another word. For instance, *Je connais tout le monde ici.* → *je connais tout notre monde.* involves the substitution of the word *le* for the word *notre*.

The longest common sub-sequence (LCS) edit distance considers only the two operations of insertion and deletion. In this setting, a substitution is a deletion followed by an insertion at the same place.

There exist well established algorithms to compute the Levenshtein edit distance, for example [28]. There also exist very fast algorithms to compute the LCS distance between two strings, like [29] or [2]. With this, the LCS edit distance between the the previous two French sentences s_1 and s_2 can be computed. Its value is $d(s_1, s_2) = 4$ because the transformation of s_1 into s_2 involves at least two deletions and two insertions.

The smaller their edit distance, the closer two sentences. Edit distances are true mathematical distances. Hence, by the axiom of separability of mathematical distances, the edit distance between two sentences is equal to 0 if and only if the two sentences are equal.

3) Sentence vector cosine: The two previous criteria to determine the similarity or the distance between sentences are purely formal. They compare words without any reference to the meaning of the words. In order to take into account semantics, we propose to use sentence vector representations. Sentence vector representations can be computed from word vector representations [21]. A simple way to compute a vector representation of a sentence is to just sum up all vector representations of the words it contains. However, this overestimates the weight of very frequent word. To avoid this we use the scheme were each word is given a weight which is proportional to its informativeness. The informativeness of a word is computed as the self-information of the word, i.e., the negative logarithm of its frequency in a given corpus.

$$I(w) = -\log f(w) \tag{4}$$

The vector representation $\overrightarrow{s}$ of a sentence s made up of the sequence of words $w_1 w_2 \ldots w_n$ is computed as in Eq. (5).

$$\overrightarrow{s} = \sum_{i=1}^{n} I(w_i) \times \overrightarrow{w_i} \tag{5}$$

With this, the similarity between two sentences s_1 and s_2 is computed as the cosine similarity between their vector representations $\cos(\overrightarrow{s_1}, \overrightarrow{s_2})$. For the two previous French sentences, the value of their similarity is 0.97.

The computation of such cosines is time-consuming. To alleviate the problem, in our implementation, given a sentence, we restrict the computation to the top 100 sentences determined by edit distance. We re-rank these 100 sentences according to their cosine similarity with respect to the given sentence.

B. Adaptation

In Nagao's model of EBMT, adaptation consists in solving an analogical equation across two languages. This is illustrated in the following French–English example.

$$\begin{array}{l}\textit{je veux re-}\\ \textit{mercier tout}\\ \textit{le monde.}\end{array} : \begin{array}{l}\textit{i want to thank}\\ \textit{everyone.}\end{array} :: \begin{array}{l}\textit{je connais tout}\\ \textit{le monde ici.}\end{array} : x \tag{6}$$

for which a possible solution is: *i know everyone here.* Let us note $A_s : A_t :: B_s : B_t$ the sentences in the order they appear in the analogical equation above. With these notations, B_s is the sentence in the source language to be translated into the target language. The task of the EBMT system is to deliver B_t, a translation of B_s. A_s and A_t are a pair of sentences in the source and the target languages which are translations of one another and which have been memorized in advance in the case base. In our setting, A_s is retrieved from B_s by using one of the three previous retrieval techniques introduced in Sect. III-A.

Symbolic techniques to solve analogical equations between strings of symbols have been proposed in previous works [26], [13], [15], [16]. All these works assume that the four strings of symbols share the same alphabet. For our setting where the four strings of symbols are four sentences, i.e., four sequences of words, this would assume that they all belong to the same language. This is not the case in Nagao's model where the analogies cross the two languages at hand. There are necessarily two different sets of words, one from each language. For this reason, we need to explore new possibilities to solve analogical equations that cross two languages.

All the previous works compare symbols (words in our setting) only by equality. It is obvious that there will be very little chance to find sentences that share a sufficient number of equal words in practice. It seems more promising to adopt a more flexible view when comparing words. For that, we capitalize on recent or past achievements in vector space models and SMT, i.e., we use monolingual distributional semantics similarity matrices and bilingual soft alignment matrices.

1) Monolingual Distributional Semantics Similarity Matrices: A way to compare two words in the same language on the level of meaning is to use distributional semantic vector space models [27]. The similarity between two words is computed as the cosine of their vector representations. For two sentences in the same language any vector space model built by any modern tool like GloVe [25] or Word2Vec [20] allows us to compute the distributional semantics similarity between any two words belonging to these two sentences.

All these similarities can be visualized in a matrix as shown with the top left and bottom right two matrices in Fig. 1. With the previous notations, these two matrices stand for the two monolingual distributional semantics similarity matrices of sentences A_s and B_s in the source language and sentences A_t and B_t in the target language.

2) Bilingual Soft Alignment Matrices: As for the bilingual comparison of two sentences which are translations of one another, sub-sentential alignment is a well-established technique in SMT [23]. It relies on the computation of association scores of translation probabilities between the words of two language, as extracted from a corpus of aligned sentences in the two languages. Various tools are available for this latter task, like Anymalign [14], SuperAlign [7] or Fast Align [5] to cite only a few. We use Anymalign [14] to get word translation association scores, because of its speed and simplicity of deployment.

Such translation associations scores can be visualized in bilingual soft alignment matrices. Two examples are shown at the bottom left and top right of Figure 1. With the previous notations (see top of Sect. III-B), these two matrices stand for the two bilingual soft alignment matrices between the translation sentences A_s and A_t on one hand and sentences B_s and B_t on the other hand.

3) Context Encoder for Analogical Equations: As mentioned above, given a sentence B_s to be translated, a sentence A_s close to B_s which has been selected by any retrieval technique and its translation A_t, the adaptation step of an EBMT system in Nagao's model consists in solving the analogical equation $A_s : A_t :: B_s : B_t$ of unknown B_t. From the given of the problem, the monolingual distributional semantics similarity matrix between A_s and B_s and the bilingual soft alignment matrix between A_s and A_t are built. In Fig. 1, they are the two top matrices. Solving the analogical equation is equivalent to guess all the values in the bilingual matrix for B_s and B_t and the monolingual matrix for B_s and B_t. In Figure 1, these are the two bottom matrices. To form the sequence of words in B_t is the job of an encoder.

We follow recent attempts at applying neural networks to this problem [10]. In particular, we use a context encoder which predicts two soft matrices from two given soft matrices [30]. A context encoder is a generative adversarial network (GAN). As any GAN, it consists of two parts a generator and a discriminator. During training, the generator is in charge of producing the matrices to fool the discriminator, and the discriminator is in charge of detecting whether the matrices the generator produced are reasonably real ones or fabricated nonsensical ones. After training, when solving analogical equations in the EBMT system, only the generator is used. The difference between a standard GAN and a context encoder lies in the loss function. In our context encoder, the loss function is a linear combination of the standard loss function used in GANs (see [8, p. 4 Algorithm 1]) and a reconstruction loss which minimizes the mena square error between the matrices guessed by the network and the ground truth. The overall effect of the loss function is to maximize the

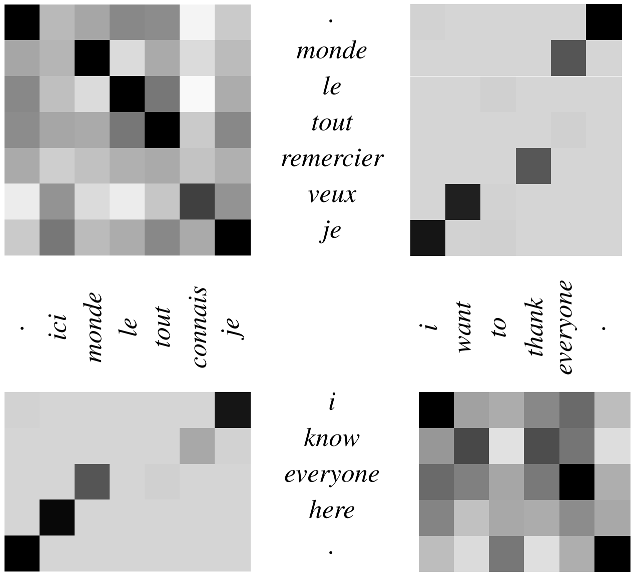

Fig. 1. Monolingual distributional semantics similarity and bilingual soft alignment matrices for an analogy in Nagao's EBMT model. Blacker cells stand for higher similarity. The matrices on the *top left* and *bottom right* are *monolingual* distributional semantics similarity matrices. The matrices on the *bottom left* and *top right* are *bilingual* sub-sentential alignment matrices.

continuity or consistency in appearance of the entire picture made up of the top and bottom matrices. This is consistent with the use of context encoders for reconstruction of images or inpainting [24]. Here, the intuition is that it ensures that the global shape of the four matrices seen as one image is basically a cross of back pixels, as can be seen in Fig. 1.

In our experiments, we use a frozen model trained on more than five thousands of analogical equations in morphology. This model outputs two matrices of values which are respectively interpreted as a bilingual matrix for B_s and B_t and a monolingual matrix for A_t and B_t. The solution decoder computes the sequence of words from the values in these two matrices. At each position in B_t, we select the word in the target language which minimizes the error in translation association score or distributional semantics similarity with the words in B_s and A_t as given by the values in the two matrices.

C. Validation

Over the last ten years, with the progress in statistical machine translation (SMT) and neural machine translation (NMT), the accuracy of EBMT, as measured by BLEU, has been largely left behind. Table I reports scores obtained by these three machine translation approaches in the WAT evaluation campaign[1] last year for different machine translation approaches from English into Japanese. This table makes it clear that the translation scores of EBMT are much lower than both SMT and NMT.

TABLE I
TRANSLATION ACCURACY AS MEASURED BY BLEU FOR DIFFERENT MACHINE TRANSLATION APPROACHES IN JAPANESE–ENGLISH IN THE WAT 2017 MACHINE TRANSLATION EVALUATION CAMPAIGN. THE BEST SCORE OF THE BEST SYSTEM FOR EACH APPROACH IS REPORTED WITH THE ABSOLUTE RANK OF THE SYSTEM AMONG ALL OTHER SYSTEMS.

MT Approach	BLEU Score	Rank
SMT	41.53	1
NMT	40.79	2
EBMT	33.06	27

The goal of this paper is to assess how far we can expect Nagao's EBMT model to perform by introducting numerical methods in the retrieval and adpatation steps. For that, we perform an oracle experiment, i.e., we cheat during the validation phase: we select the sentences using their BLEU score against the reference translations. In an actual setting, this is of course impossible as the reference translations are unknown.

For one sentence to be translated, we retrieve a certain number of similar examples and their translation from bilingual corpus using the three techniques described in Sect. III-A. In addition, we also consider a hybrid system. It takes the best translation candidate from the three previous retrieval techniques and keeps the best one. The overall average BLEU score is then computed on these best translation candidates.

IV. EXPERIMENTS

A. Data

We choose French-English as the language pair. It is a classical language pair for MT, not so simple, and not so difficult, for MT. Reasonably high scores are usually obtained. We use Word2Vec to pre-train word vector space models in which we compute the distributional semantics similarity by

[1] Workshop on Asian Translation, http://lotus.kuee.kyoto-u.ac.jp/WAT/. Figures in Table I are copied from the page /evaluation/list.php?t=1&o=1 with Juman [19] as the Japanese segmenter.

TABLE II
TRAINING WORD VECTOR MODELS

Languages	French, English
Data	WikiDumps (Version.20170601)
Tokens	100 million
Model	Word2vec (CBOW model)
Dimensions	300
Window size	7
Minimun count	5

TABLE III
STATISTICS ON THE FRENCH-ENGLISH DATA OF THE TATOEBA CORPUS (SENTENCES OF LENGTH LESS THAN 10 TOKENS)

	French		English	
	Training	Test	Training	Test
Lines	109,390	1,200	109,390	1,200
Tokens	753,880	8,230	725,203	7,973
Types	3,924,245	42,845	3,305,809	36,256

classical cosine similarity. The training corpora are Wikidumps from the 2017.06.01. Table IV-A gives the parameter setting.

We use the Tatoeba Corpus[2] as our bilingual corpus. It indeed fits EBMT because it is made up of repetitive sentences which exhibit large similarities. We retain sentences of less than 10 words in length and select 90 % of them for training and the other 10 % for testing. We use the training part to estimate word translation association scores using Anymalign. Table III gives statistics on these data.

B. Results

We assess the three different techniques introduced in Sect. III-A to find the N most similar sentences in an oracle setting as described in Sect. III-C. Experiments are conducted fro different values of N. The results are shown in Table IV. We also compare our EBMT systems with two baseline systems using standard technology: an SMT system (GIZA++, Moses, KenLM, MERT) and an NMT system (OpenNMT). In addition, we build a hybrid system which uses the three techniques and retains the best candidates.

1) Comparison of different retrieval techniques and MT approaches: Compared to nowadays standard evaluation data sets which offer millions of aligned sentences, our data set is relatively small. It is known that, on less data, neural machine translation does not perform as well as statistical machine translation [11]. Our experiment confirms this: the score of the NMT system is lower than the score of the SMT system by 5 BLEU points. Now, the scores of our oracle experiments show that it is possible to reach a score between a baseline SMT and a baseline NMT system using an EBMT system.

The surprising result is that the best retrieval technique is Dice coefficient. This may mean that the highest number of exact words is an important criterion to produce useful matrices for the adaptation step. Without surprise, the hybrid system would consistently outperform each of the individual three techniques it uses.

2) Influence of the number of retrieved examples: The second experiment tests the influence of N when allowing the system to use the top N retrieved examples. As already reported for another EBMT system [17] which also uses analogy as its core technique, the results show that using the top first retrieved example does not necessary lead to the best translation accuracy.

The main positive result is that when the value of N reaches a certain value, our results can exceed those of NMT: 10 top sentences for Dice coefficient and edit distances. Cosine would need 40 top sentences. The hybrid system would be able to beat both the NMT and the SMT baselines if it were able to select the best translation using the top 10 retrieved examples. This shows that EBMT is definitely a promising technique for smaller data sets.

V. CONCLUSION

We introduced numerical methods in Nagao's EBMT model, precisely in the retrieval and adaptation steps according to the description of EBMT as CBR. As for retrieval, we proposed and assessed three different techniques to measure similarity. As for adaptation, we introduced a neural network-based technique to solve analogies between sentences that cross two languages. This was made possible by using distributional semantics similarity from vector space models and translation association scores from SMT.

The main positive result is that when the value of N reaches a certain value, our results can exceed those of an NMT baseline. Using a hybrid system would even beat an SMT baseline. However, there still exist bottlenecks. We list up only two hereafter.

Firstly, our analogy solver makes a hypothesis on the length of the solution, which is essentially wrong. It assumes that the difference in lengths in the source language is equal to the difference in length in the target language. The use of a sentence length model like the one proposed to align sentences from translated texts [6] should directly lead to a better translation accuracy.

Secondly, the frozen context encoder model we used to solve analogical equations between sentences did not match our task. It was learned from data in morphology and as such did not fit our data, which are of syntactic nature by essence. Building a better fitted frozen model should also lead to better a translation accuracy.

ACKNOWLEDGEMENTS

This research was supported in part by the Japanese Society for the Promotion of Science (JSPS) under grant 18K11447 entitled: Self-explainable and fast-to-train example-based machine translation using neural networks.

[2] http://www.manythings.org/anki/

TABLE IV
BLEU SCORES FOR DIFFERENT VALUES OF N FOR TOP N RETRIEVAL AND FOR THE THREE DIFFERENT RETRIEVAL METHODS. BOLDFACE SCORES ARE THE BEST ONES ON EACH ROW FOR OUR SYSTEMS

N		1	10	20	30	40	50	60	70	80	90	100
Dice coefficient		44.70	61.87	62.38	62.75	62.98	63.13	63.30	63.47	63.64	63.79	**63.83**
Edit distance		32.67	60.13	61.69	62.30	62.65	63.02	63.21	63.41	63.52	63.52	**63.58**
Sentence vector cosine		36.44	54.39	57.82	59.58	60.84	61.90	62.83	63.20	63.42	63.50	**63.58**
Hybrid system		52.90	**65.58**	64.22	64.56	64.72	64.87	64.98	65.09	65.17	65.29	65.37
SMT baseline	65.23											
NMT baseline	59.87											

REFERENCES

[1] A. Aamodt and E. Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[2] L. Allison and T. I. Dix. A bit string longest common subsequence algorithm. *Information Processing Letters*, 23:305–310, 1986.

[3] B. Collins and H. Somers. *Recent Advances in Example-Based Machine Translation*, chapter EBMT seen as case-based reasoning, pages 115–153. Springer Netherlands, Dordrecht, 2003.

[4] E. Denoual. Analogical translation of unknown words in a statistical machine translation framework. In *Proceedings of the Eleventh Machine Translation Summit (MT Summit XI)*, pages 135–141, Copenhagen, Denmark, 2007.

[5] C. Dyer, V. Chahuneau, and N. A. Smith. A simple, fast, and effective re-parameterization of IBM model 2. In *Proceedings of NAACL-HLT 2013*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[6] W. A. Gale and K. W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):76–102, 1993.

[7] C.-L. Goh and E. Sumita. A feature-rich supervised word alignment model for phrase-based statistical machine translation. *International Journal on Asian Language Processing*, 19(3):109–125, 2009.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] P. Jaccard. Lois de distribution florale dans la zone alpine. *Bulletin de la Société Vaudoise de Sciences Naturelles*, 38(144):69–130, 1901.

[10] V. Kaveeta and Y. Lepage. Solving analogical equations between strings of symbols using neural networks. In *Proceedings of ICCBR-16 Workshops*, pages 67–76, 2016.

[11] P. Koehn and R. Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, Canada, 2017. Association for Computational Linguistics.

[12] P. Langlais. Mapping source to target strings without alignment by analogical learning: A case study with transliteration. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 684–689, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[13] P. Langlais, P. Zweigenbaum, and F. Yvon. Improvements in analogical learning: application to translating multi-terms of the medical domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 487–495, Athens, Greece, March 2009. Association for Computational Linguistics.

[14] A. Lardilleux and Y. Lepage. Sampling-based multilingual alignment. In *Recent Advances in Natural Language Processing, (RANLP 2009)*, pages 214–218, Borovets, Bulgaria, September 2009.

[15] Y. Lepage. Solving analogies on words: an algorithm. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL 1998)*, volume 1, pages 728–734. Association for Computational Linguistics, 1998.

[16] Y. Lepage. Character–position arithmetic for analogy questions between word forms. In *Proceedings of the Computational Analogy Workshop at the 24th International Conference on Case-Based Reasoning (ICCBR-17)*, pages 17–26, Trondheim, Norway, August 2017.

[17] Y. Lepage and E. Denoual. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, 19(3-4):251–282, 2005.

[18] Y. Lepage and J. Lieber. Case-based translation: First steps from a knowledge-light approach based on analogy to a knowledge-intensive one. In M. T. Cox, P. Funk, and S. Begum, editors, *Proceedings of the 26th International Conference on Case-Based Reasoning (ICCBR-18)*, pages 273–288, Stockholm, Sweden, August 2018. Springer.

[19] Y. Matsumoto, S. Kurohashi, Y. Nyoki, H. Shinho, and M. Nagao. User's guide for the Juman system, a user-extensible morphological analyzer for Japanese (version 0.5). Technical report, Kurohashi and Kawahara Lab, Kyoto University, 1997.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations (ICLR 2013)*, 2013.

[21] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*, volume 13, pages 746–751, 2013.

[22] M. Nagao. A framework of a mechanical translation between japanese and english by analogy principle. *Artificial and human intelligence*, pages 351–354, 1984.

[23] F. J. Och and H. Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.

[24] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[25] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, volume 14, pages 1532–1543, 2014.

[26] R. Rhouma and P. Langlais. Experiments in learning to solve formal analogical equations. In M. T. Cox, P. Funk, and S. Begum, editors, *Proceedings of the 26th International Conference on Case-Based Reasoning (ICCBR-18)*, pages 438–453, Stockholm, Sweden, August 2018. Springer.

[27] P. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[28] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173, 1974.

[29] S. Wu, U. Manber, G. Myers, and W. Miller. An O(*NP*) sequence comparison algorithm. *Information Processing Letters*, 35:317–323, April 1990.

[30] T. Zhao and Y. Lepage. Context encoder for analogies on strings [submitted]. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation (PACLIC 32)*, pages ??–??, 2018.