

# An Approach to Case-Based Reasoning Based on Local Enrichment of the Case Base

Yves Lepage and Jean Lieber\*

Waseda University, IPS, 2-7 Hibikino, 808-0135 Kitakyushu, Japan

yves.lepage@waseda.jp

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

jean.lieber@loria.fr

**Abstract.** This paper describes an approach to case-based reasoning by which the case base is enriched at reasoning time. Enrichment results from the local application of variations to seed cases: new hypothetical cases are created which get closer and closer to the target problem. The creation of these hypothetical cases is based on structures associated to the problem and solution spaces, called variation spaces, that enable defining a language of adaptation rules. Ultimately reaching the target problem (exactly or nearly) allows the system to deliver a solution. Application of the proposed approach to machine translation shows behind state-of-the-art, but promising results.

**Keywords:** analogical reasoning, case base enrichment, case-based machine translation, case-based reasoning

## 1 Introduction

In machine learning, some techniques are used to enrich the training set in order to improve the accuracy of a learning system. This is called data augmentation. It can be done using general transformations (flipping, rotating, etc. images when this does not affect the class in image classification, see for example Taylor and Nitschke [17]), adding some controlled noise (like Gaussian noise on images, see Hussain et al. [7]), or by analogical reasoning (see, e.g. Couceiro et al. [2]).

While case-based reasoning (CBR, Richter and Weber [16]) usually requires less data than most current machine learning techniques, the enrichment of the case base can be useful. In this paper, we propose to perform the enrichment of the case base in a “case-based way”: the case base is enriched with new source cases that are “around” the target problem. This constitutes a local enrichment of the case base, computed on-line, at the time of problem-solving. The case base is enriched with the help of learned adaptation rules: given a source case, another source case closer to the target problem is generated thanks to such a

---

\* The authors want to thank the anonymous reviewers for their detailed comments. They have tried to do their best to take them into account. The first author is supported by JSPS Grant-In-Aid 18K11447: “Self-explainable and fast-to-train example-based machine translation using neural networks.”

rule. The process can be repeated, though each application of a rule may degrade the generated case, in the sense that it is less and less likely to be a licit case.

To deal with this issue, the notion of penalized case is introduced: this is a hypothetical case, whose likelihood to be licit is characterized by a number that penalizes the re-usability of the case. Therefore, the enriched case base is a set of penalized cases: the cases to be reused are the ones which offer the best compromise between the similarity to the target problem and their penalties.

Section 2 presents general definitions. The notions of penalized cases are presented in Section 3. The notions of variations between problems and between solutions are defined in Section 4, with some strong assumptions that make the general ideas easier to understand. Section 5 is the core of the paper: it presents the approach for local enrichment of the case base. It presents a first approach consisting in generating from the case base all the penalized cases with a penalty under a given threshold. Since this approach leads to a combinatorial explosion of the case base, a more practical approach is presented afterwards. They show how case-based translation (Lepage and Lieber [13]) can be performed thanks to the approach of local enrichment of the case base. The results are demonstrative and promising.

## 2 Preliminaries

### 2.1 Notations and assumptions on case-based reasoning

Let  $\mathcal{P}$  and  $\mathcal{S}$  be two given sets, called the *problem space* and the *solution space*. A *problem* is by definition an element  $x$  of  $\mathcal{P}$  and a *solution*, an element  $y$  of  $\mathcal{S}$ . A relation  $\rightsquigarrow$  on  $\mathcal{P} \times \mathcal{S}$  is assumed to exist and  $x \rightsquigarrow y$  is read “ $x$  has for solution  $y$ ” or “ $y$  is a solution to  $x$ ”. A case is a pair  $(x, y) \in \mathcal{P} \times \mathcal{S}$  such that  $x \rightsquigarrow y$ . The *case base*  $\text{CB}$  is a finite and nonempty set of cases. A *source case* is an element  $(x^s, y^s)$  of  $\text{CB}$ . CBR aims at solving a new problem  $x^t$  called the *target problem* with the help of the case base. It usually consists of the following steps:

- Retrieval (aka “retrieve” in Aamodt and Plaza [1]) selects a subset of  $\text{CB}$ ;
- Adaptation (aka “reuse”) proposes a *plausible* solution  $y^t$  to  $x^t$ , using the retrieved source cases;
- Learning (aka as “revise” and “retain”) consists in validating/correcting  $y^t$  (for example, with the help of a human expert) and in storing the newly formed case  $(x^t, y^t)$  in the case base, if this storage is judged appropriate.

It is worth noting that, for many applications,  $y^t$  is only a plausible solution: CBR often functions as a hypothetical reasoning process whose use is motivated by the incompleteness of the knowledge of the relation  $\rightsquigarrow$ . The term “hypothetical case” stands for any pair  $(x, y) \in \mathcal{P} \times \mathcal{S}$ , though this notion is generally used when a solution  $y$  to  $x$  is plausibly inferred. By contrast, if  $(x, y)$  is a case —and thus,  $x \rightsquigarrow y$ — it is called a *licit case* to stress its certainty.

In some applications, other knowledge containers are used (Richter and Weber [16]): the domain knowledge, the retrieval knowledge, and the adaptation knowledge ( $\text{AK}$ ). These four knowledge containers are interrelated. In particular, there are studies on learning  $\text{AK}$  using  $\text{CB}$  presented further.

## 2.2 Analogies

An analogy is a quaternary relation on a set  $\mathcal{U}$  denoted by  $A : B :: C : D$  for  $(A, B, C, D) \in \mathcal{U}^4$ , that is to be read “ $A$  is to  $B$  as  $C$  is to  $D$ ”. It satisfies the following postulates (for any  $(A, B, C, D) \in \mathcal{U}^4$ ):<sup>1</sup>

**(reflexivity of conformity)**  $A : B :: A : B$ ;

**(symmetry of conformity)** if  $A : B :: C : D$  then  $C : D :: A : B$ ;

**(exchange of the means)** if  $A : B :: C : D$  then  $A : C :: B : D$ .

A *ratio* is an expression of the form  $P : Q$  (“ $P$  is to  $Q$ ”), the relation  $::$  (“as”) is called conformity. Thus, analogy is a conformity of ratios.<sup>2</sup>

Classical examples of analogies are as follows:

**(geometrical analogy)** Here the ratio is division, conformity is equality, and  $\mathcal{U} = \mathbb{R} \setminus \{0\}$ . This analogy is defined by  $A : B :: C : D$  if  $\frac{A}{B} = \frac{C}{D}$ .

**(arithmetic analogy)** Here the ratio is subtraction, conformity is equality, and  $\mathcal{U} = \mathbb{R}$ . This analogy is defined by  $A : B :: C : D$  if  $A - B = C - D$ .

**(analogy on tuples)** If an analogy is defined on each set  $\mathcal{U}_i$  ( $1 \leq i \leq n$ ) and  $\mathcal{U} = \mathcal{U}_1 \times \mathcal{U}_2 \times \dots \times \mathcal{U}_n$  then the following analogy can be defined on  $\mathcal{U}$ :  
 $A : B :: C : D$  if for every  $i$ ,  $A_i : B_i :: C_i : D_i$ .

**(analogies on strings)** Let  $\text{dist}$  be the LCS distance.<sup>3</sup> The Parikh vector of a string is the tuple of the number of occurrences of each character in the string. A ratio  $P : Q$  between two strings  $P$  and  $Q$  can be defined as the difference of their Parikh vectors (arithmetic analogies on tuples) plus the LCS distance between them. However, this ratio does not entail the exchange of the means because  $\text{dist}(A, B) = \text{dist}(C, D)$  does not imply  $\text{dist}(A, C) = \text{dist}(B, D)$  in general. To define an analogy it is necessary to state:  $A : B :: C : D$  if  $A : B = C : D$  and  $A : C = B : D$ .

An analogical equation is an expression of the form  $A : B :: C : y$ , where  $y$  is the unknown. Solving it amounts to find the set of  $y$  such that the analogy holds. It may have 0, 1 or several solutions, depending on the type of analogies: for geometrical analogies on  $\mathbb{R} \setminus \{0\}$  and arithmetic analogies on  $\mathbb{R}$ , every analogical equation has exactly one solution. By contrast, with analogies on strings as defined above, an analogical equation may have 0, 1 or several solutions.

<sup>1</sup> Some authors consider that analogy requires additional postulates (Lepage [10]). However, only these three postulates are used in this paper.

<sup>2</sup> When conformity is an equivalence relation, reflexivity and symmetry are straightforward. But conformity is not necessarily an equivalence relation.

<sup>3</sup> The LCS (“longest common subsequence”) distance is an edit distance based on the character insertion and character deletion edit operations, with a cost of 1 for both. In other terms, if  $P$  and  $Q$  are two strings and  $L$  is the LCS of  $P$  and  $Q$ , then  $\text{dist}(P, Q) = (|P| - |L|) + (|Q| - |L|)$ .

### 3 Penalized cases and penalized case bases

The enrichment of the case base presented here is based on hypothetical reasoning: it generates hypothetical cases  $(\mathbf{x}, \mathbf{y})$ , i.e., the assertion “ $\mathbf{x} \rightsquigarrow \mathbf{y}$ ” is uncertain. Thus, a hypothetical case is less trustworthy than a licit case (such as a source case), and thus, the former has to be penalized in the reasoning. Two hypothetical cases have different penalties if one of them is more uncertain than the other. The notion of penalty as a way to model uncertainty is introduced for this purpose. The penalties are associated to hypothetical inferences: the more uncertain the inference, the higher the inference cost, that is the additional penalty associated with the inference. Finally, the notion of penalized case is introduced: they are triples  $(\mathbf{x}, \mathbf{y}, \pi)$  where  $\mathbf{x} \in \mathcal{P}$ ,  $\mathbf{y} \in \mathcal{S}$  and  $\pi$  is a penalty. An enriched case base is actually a set of penalized cases, i.e., a penalized case base.

#### 3.1 Uncertainty and Penalties

Hypothetical reasoning leads to uncertain results. In this paper, uncertainty of an event is measured by an *uncertainty penalty* (or, simply, a *penalty*)  $\pi \in [0, \infty]$  such that the higher  $\pi$  is, the less certain the event is. The penalty of an event that is certain is  $\pi = 0$ . The penalty of an impossible event is  $\pi = \infty$ .<sup>4</sup>

**Remark 1:** If two penalties  $\pi^1$  and  $\pi^2$  are associated to the same event, with  $\pi^1 < \pi^2$ , the lower penalty —associated to the higher certainty— is kept. In other words, if  $\pi$  is associated to an event  $e$ , then every  $\pi' \geq \pi$  can also be associated to  $e$ .

#### 3.2 Cost associated to a hypothetical inference

Let  $\varphi^0$  be a piece of knowledge whose uncertainty (to be consistent with the real world) has an uncertainty penalty  $\pi^0$ . From  $\varphi^0$ , a new piece of knowledge  $\varphi^1$  can be produced by a hypothetical inference **hypo**. Since **hypo** is hypothetical, it adds some uncertainty, thus an uncertainty penalty  $\pi^1$  associated to  $\varphi^1$  can be computed as  $\pi^1 = \pi^0 + c$  where  $c > 0$  measures the additional uncertainty of **hypo**.  $c$  is assumed to be computed on the basis of **hypo** by a function called **cost**:  $c = \text{cost}(\text{hypo})$ .<sup>5</sup> In summary:

$$\begin{array}{l} \text{if } \pi^0 \text{ is an uncertainty penalty associated to } \varphi^0 \\ \text{and } \varphi^1 \text{ is inferred from } \varphi^0 \text{ by the hypothetical inference } \text{hypo} \\ \text{then } \pi^1 = \pi^0 + \text{cost}(\text{hypo}) \text{ is an uncertainty penalty associated to } \varphi^1 \end{array} \quad (1)$$

<sup>4</sup> If uncertainty is modeled thanks to a probability measure, it is possible to associate to a probability  $P \in [0; 1]$  a penalty  $\pi = -\log P \in [0; \infty]$ . If uncertainty is thought of as a measure of the gap to consistency with the real world, it is possible to associate a distance to it. Then, by definition, licit cases have a penalty of 0. The representation of uncertainty by penalties is chosen in this paper for generality of expression.

<sup>5</sup> Once again, costs could be associated to probabilities: **cost(hypo)** could be defined by  $-\log P(\varphi^2 \mid \varphi^1)$ , where  $P(\varphi^2 \mid \varphi^1)$  is the probability of  $\varphi^2$  being true given that  $\varphi^1$  is. But they can also be associated to distance. **cost(hypo)** could be defined as  $\text{dist}(\varphi^1, \varphi^2)$  which expresses the additional uncertainty on  $\varphi^2$  when inferred from  $\varphi^1$ .

Now, if  $\varphi^0, \varphi^1, \dots, \varphi^n$  are pieces of knowledge,  $\varphi^0$  being certain (it can be associated to an uncertainty penalty  $\pi^0 = 0$ ) and  $\varphi^i$  being produced by a hypothetical inference  $\text{hypo}_i$  ( $1 \leq i \leq n$ ), then, according to (1), the uncertainty penalty  $\pi^n = \sum_{i=1}^n \text{cost}(\text{hypo}_i)$  can be associated to  $\varphi^n$ .

In particular, let us consider the simple approach to CBR consisting in retrieving a source case  $(\mathbf{x}^s, \mathbf{y}^s)$  and in reusing (without modification)  $\mathbf{y}^s$  as a plausible solution to the target problem  $\mathbf{x}^t$  ( $\mathbf{y}^t = \mathbf{y}^s$ ). If retrieval is based on a distance function  $\text{dist}$ , then it is reasonable to assume that the higher  $\text{dist}(\mathbf{x}^s, \mathbf{x}^t)$  is, the more uncertain the assertion  $\mathbf{x}^t \rightsquigarrow \mathbf{y}^t$  is, and the higher the cost of this inference. For this reason,  $\text{dist}(\mathbf{x}^s, \mathbf{x}^t)$  can be used to measure the cost of this inference: this is how the distance function is interpreted and used in the rest of the paper.

**Remark 2:** If an uncertain piece of knowledge  $\varphi$  is inferred by two hypothetical inferences, leading to two penalty values  $\pi^1$  and  $\pi^2$ , then, following Remark 1,  $\varphi$  is associated to the penalty  $\min(\pi^1, \pi^2)$ .

### 3.3 Penalized cases

In this paper, every hypothetical case  $(\mathbf{x}, \mathbf{y}) \in \mathcal{P} \times \mathcal{S}$  is either a licit case ( $\mathbf{x} \rightsquigarrow \mathbf{y}$ ) or not ( $\mathbf{x} \not\rightsquigarrow \mathbf{y}$ ): there is no gradual distinction between licit and illicit cases. By contrast, a hypothetical case is more or less certain to be a licit case. So, hypothetical cases should be preferred on the basis of their respective chances of being licit. A *penalized case* is a triple  $(\mathbf{x}, \mathbf{y}, \pi)$  with  $(\mathbf{x}, \mathbf{y})$ , a hypothetical case and  $\pi$ , a penalty measuring the uncertainty that  $\mathbf{x} \rightsquigarrow \mathbf{y}$ . The estimation of  $\pi$  is made on the basis of an inference that was applied to generate the hypothetical case  $(\mathbf{x}, \mathbf{y})$ . If  $\pi = 0$ , then  $(\mathbf{x}, \mathbf{y}, \pi) = (\mathbf{x}, \mathbf{y}, 0)$  is assimilated to the case  $(\mathbf{x}, \mathbf{y})$ .

A *penalized case base* PCB is a finite set of penalized cases  $(\mathbf{x}^s, \mathbf{y}^s, \pi^s) \in \mathcal{P} \times \mathcal{S} \times [0, \infty]$ . In particular, CB is a penalized case base with penalties set to 0.

When a penalized case base PCB is used, instead of a classical case base, how does it affect the CBR process? An answer is to take into account the penalties of the case by adding them to the cost of the inference. For example, since  $\text{dist}$  is interpreted as a cost of the simple CBR inference based on the retrieval of a single source case and reusing it as such (cf. Section 3.2), this approach to CBR consists in selecting the  $(\mathbf{x}^s, \mathbf{y}^s, \pi^s) \in \text{PCB}$  which minimizes  $\text{dist}(\mathbf{x}^s, \mathbf{x}^t) + \pi^s$ .

## 4 Problem and solution variations

### 4.1 Definitions

Intuitively, the variation from a problem  $\mathbf{x}^i$  to a problem  $\mathbf{x}^j$ , denoted by  $\overrightarrow{\mathbf{x}^i \mathbf{x}^j}$  in this paper, encodes the information necessary to transform  $\mathbf{x}^i$  into  $\mathbf{x}^j$ . More formally, a triple  $(\Delta\mathcal{P}, +, \overrightarrow{\cdot})$  called the *problem variation space* and satisfying the following postulates is assumed to exist:

- (i)  $(\Delta\mathcal{P}, +)$  is a commutative group. Its identity element is denoted by  $\vec{0}$ ; the inverse of  $\vec{u} \in \Delta\mathcal{P}$  is denoted by  $-\vec{u}$ .<sup>6</sup>
- (ii)  $\vec{\cdot}$  is an onto mapping  $(\mathbf{x}^i, \mathbf{x}^j) \in \mathcal{P}^2 \mapsto \overrightarrow{\mathbf{x}^i \mathbf{x}^j} \in \Delta\mathcal{P}$ : for each  $\vec{u} \in \Delta\mathcal{P}$ , there exists  $(\mathbf{x}^i, \mathbf{x}^j) \in \mathcal{P}^2$  such that  $\vec{u} = \overrightarrow{\mathbf{x}^i \mathbf{x}^j}$ .
- (iii) For every  $\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^k \in \mathcal{P}$ ,  $\overrightarrow{\mathbf{x}^i \mathbf{x}^j} + \overrightarrow{\mathbf{x}^j \mathbf{x}^k} = \overrightarrow{\mathbf{x}^i \mathbf{x}^k}$ .
- (iv) For each  $\mathbf{x}^i \in \mathcal{P}$  and  $\vec{u} \in \Delta\mathcal{P}$ , there exists *at most* one  $\mathbf{x}^j \in \mathcal{P}$  such that  $\overrightarrow{\mathbf{x}^i \mathbf{x}^j} = \vec{u}$ . This  $\mathbf{x}^j$ , *when it exists*, is denoted by  $tr_{\vec{u}}(\mathbf{x}^i)$  (*tr* stands for “translation”, borrowing the term from the field of vector spaces).

From these postulates, the following properties can be deduced:

$$\overrightarrow{\mathbf{x}^i \mathbf{x}^j} = \vec{0} \quad \text{iff} \quad \mathbf{x}^i = \mathbf{x}^j \quad (2)$$

$$-\overrightarrow{\mathbf{x}^i \mathbf{x}^j} = \overrightarrow{\mathbf{x}^j \mathbf{x}^i} \quad (3)$$

$$\text{if } \overrightarrow{\mathbf{x}^1 \mathbf{x}^2} = \overrightarrow{\mathbf{x}^3 \mathbf{x}^4} \text{ then } \overrightarrow{\mathbf{x}^1 \mathbf{x}^3} = \overrightarrow{\mathbf{x}^2 \mathbf{x}^4} \quad (4)$$

for every  $\mathbf{x}^i, \mathbf{x}^j, \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4 \in \mathcal{P}$ .

There are many ways of defining  $\Delta\mathcal{P}$  and the mapping  $\vec{\cdot}$ . They depend partly upon the problem space  $\mathcal{P}$ . For example:

- If  $\mathcal{P}$  is an affine space of dimension  $n$  on  $\mathbb{R}$ ,  $\Delta\mathcal{P}$  can be the vector space  $\mathbb{R}^n$  associated with  $\mathcal{P}$ :  $\overrightarrow{\mathbf{x}^i \mathbf{x}^j} = (\mathbf{x}_1^j - \mathbf{x}_1^i, \mathbf{x}_2^j - \mathbf{x}_2^i, \dots, \mathbf{x}_n^j - \mathbf{x}_n^i)$  and  $tr_{\vec{u}}$  is the translation operator of vector  $\vec{u}$ . This example explains the notations chosen in this paper.
- More generally, if  $\mathcal{P}$  is defined by attribute-value pairs, the problem of defining  $\overrightarrow{\mathbf{x}^i \mathbf{x}^j}$  can be reduced to the problem of defining the variation from  $\mathbf{x}^i$  to  $\mathbf{x}^j$  for each attribute. This is considered in particular in d’Aquin et al. [4].

In the same way and with the same notations, a solution variation space  $(\Delta\mathcal{S}, +, \vec{\cdot})$  can be defined.

## 4.2 Adaptation knowledge learning expressed in terms of variations

The seminal paper of Hanney and Keane [6] presents the main principles of the AK learning issue. They are reformulated below, thanks to the notions of variations introduced above.

From the case base, the multiset<sup>7</sup>

$$\text{TS} = \left\{ \left\{ \left( \overrightarrow{\mathbf{x}^i \mathbf{x}^j}, \overrightarrow{\mathbf{y}^i \mathbf{y}^j} \right) \mid (\mathbf{x}^i, \mathbf{y}^i), (\mathbf{x}^j, \mathbf{y}^j) \in \text{CB}, \text{ with } \mathbf{x}^i \neq \mathbf{x}^j \right\} \right\}$$

<sup>6</sup>  $(\Delta\mathcal{P}, +)$  being a commutative group means that  $\Delta\mathcal{P}$  is a set, that  $+$  is an associative and commutative operation on  $\Delta\mathcal{P}$ , and that every  $\vec{u} \in \Delta\mathcal{P}$  has an inverse element  $-\vec{u}$  (meaning  $\vec{u} + (-\vec{u}) = \vec{0}$ ).

<sup>7</sup> A multiset is denoted with double braces; for example  $M = \{\{a, a, b, c, c, c\}\}$  contains  $a$  with multiplicity 2,  $b$  with multiplicity 1 and  $c$  with multiplicity 3. Thus the cardinality of  $M$  is  $|M| = 2 + 1 + 3 = 6$ .

is computed. This multiset is used in the training of a supervised learning process (the inputs of the examples are the  $\overrightarrow{\mathbf{x}^i \mathbf{x}^j}$ , the outputs are the  $\overrightarrow{\mathbf{y}^i \mathbf{y}^j}$ ). The learned model is used as adaptation knowledge.

Several studies have followed this scheme, and a few examples are given below. In Craw et al. [3], a variety of learning techniques are used, in particular decision tree induction and ensemble learning. In d'Aquin et al. [4], frequent closed itemset extraction is used. The expert interpretation enables to produce adaptation rules to be added to AK. In Jalali et al. [8], an ensemble approach provides adaptation rules for a nominal representation (feature-value pairs, where values are categories).

An example of adaptation learning approach suited for discrete representations is as follows. First, a triple  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}}, \mathbf{c}) \in \Delta\mathcal{P} \times \Delta\mathcal{S} \times [0, \infty[$  such that  $\overrightarrow{\mathbf{u}} \neq \overrightarrow{\mathbf{0}}$  can be seen as an adaptation rule (for  $(\mathbf{x}, \mathbf{y}, \pi)$ , a penalized case and  $\mathbf{x}^t \in \mathcal{P}$ ):

**if**  $\overrightarrow{\mathbf{x}\mathbf{x}^t} = \overrightarrow{\mathbf{u}}$  **then**  $\mathbf{y}^t = tr_{\overrightarrow{\mathbf{v}}}(\mathbf{y})$  is a plausible solution of  $\mathbf{x}^t$ , with penalty  $\pi + \mathbf{c}$

(recall that  $\mathbf{y}^t = tr_{\overrightarrow{\mathbf{v}}}(\mathbf{y})$  iff  $\overrightarrow{\mathbf{y}\mathbf{y}^t} = \overrightarrow{\mathbf{v}}$ ). Among the  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}}, \mathbf{c}) \in \Delta\mathcal{P} \times \Delta\mathcal{S} \times [0, \infty[$  ( $\overrightarrow{\mathbf{u}} \neq \overrightarrow{\mathbf{0}}$ ), the ones that are selected are the ones which are the most supported by the training set. More formally, let  $\mathbf{supp}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}})$  (the *support* of the ordered pair  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}})$ ) be the multiplicity of  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}})$  in TS. In other terms:

$$\mathbf{supp}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}}) = \left| \left\{ ((\mathbf{x}^i, \mathbf{y}^i), (\mathbf{x}^j, \mathbf{y}^j)) \in \mathbf{CB}^2 \mid \overrightarrow{\mathbf{x}^i \mathbf{x}^j} = \overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{y}^i \mathbf{y}^j} = \overrightarrow{\mathbf{v}} \right\} \right|$$

Hence, the adaptation knowledge learning process consists in computing the pairs  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}})$  such that their support is above a given threshold  $\tau_{\mathbf{supp}}$ . It is assumed here that  $\tau_{\mathbf{supp}} \geq 2$ . The value of the support is used on the basis of the following heuristics: the higher the support is, the less the application of the adaptation rule adds uncertainty. Therefore, a value  $\mathbf{c}$  is computed thanks to a decreasing function  $f : \mathbb{N} \setminus \{0, 1\} \rightarrow \mathbb{R}$  by  $\mathbf{c} = f(\mathbf{supp}(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}}))$ . For our experiments, we have chosen  $f(n) = 1/n$ . Finally, the rule  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}}, \mathbf{c})$  is added to AK.

With this adaptation knowledge learning approach, it is noteworthy that for each learned adaptation rule  $(\overrightarrow{\mathbf{u}}, \overrightarrow{\mathbf{v}}, \mathbf{c})$  there is another learned adaptation rule  $(-\overrightarrow{\mathbf{u}}, -\overrightarrow{\mathbf{v}}, \mathbf{c})$ . Indeed,  $(\overrightarrow{\mathbf{x}^i \mathbf{x}^j}, \overrightarrow{\mathbf{y}^i \mathbf{y}^j})$  occurs in the multiset TS with the same multiplicity as  $(\overrightarrow{\mathbf{x}^j \mathbf{x}^i}, \overrightarrow{\mathbf{y}^j \mathbf{y}^i})$ .

### 4.3 Variation-based analogies

Let us consider the relation on problems defined, for  $A, B, C, D \in \mathcal{P}$ , as follows:

$$A : B :: C : D \quad \text{if} \quad \overrightarrow{AB} = \overrightarrow{CD}$$

It satisfies the postulates of analogy. An analogy on  $\mathcal{S}$  can be defined likewise.

Therefore, using the problem variation space and the solution variation space, an analogy on  $\mathcal{P}$  and an analogy on  $\mathcal{S}$  can be built, and thus, the approach to CBR based on the following principle (called *extrapolation* in Lieber et al. [14] and used in Lepage and Denoual [12] and Lepage and Lieber [13]) can be applied:

- Given a target problem  $\mathbf{x}^t$ , a triple  $((\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), (\mathbf{x}^3, \mathbf{y}^3)) \in \mathbf{CB}^3$  is retrieved such that  $\mathbf{x}^1 : \mathbf{x}^2 :: \mathbf{x}^3 : \mathbf{x}^t$  in the problem space (i.e.,  $\overrightarrow{\mathbf{x}^1 \mathbf{x}^2} = \overrightarrow{\mathbf{x}^3 \mathbf{x}^t}$ ).
- Then, the analogical equation  $\mathbf{y}^1 : \mathbf{y}^2 :: \mathbf{y}^3 : y$  in the solution space is solved, and the solution of this equation is given as a plausible solution to  $\mathbf{x}^t$  (in the framework of the postulates given below, this solution, when it exists, is unique, and verifies  $y = tr_{\overrightarrow{v}}(\mathbf{y}^3)$  with  $\overrightarrow{v} = \overrightarrow{\mathbf{y}^1 \mathbf{y}^2}$ ).

#### 4.4 Analogy-based variations

In the domain of strings, it is possible to define a vector corresponding to a ratio as follows (using the notion of ratios introduced in Section 2.2):  $\overrightarrow{AB} = A : B$ . However, as mentioned in Section 2.2, the definition of a ratio alone does not make an analogy. In such a domain, we impose for  $(A, B, C, D) \in \mathcal{P}^4$ :

$$\overrightarrow{AB} = \overrightarrow{CD} \quad \text{if} \quad A : B :: C : D, \text{ i.e., if } A : B = C : D \text{ and } A : C = B : D$$

In other words, in such a domain, we implement the extrapolation approach mentioned at the end of Section 4.3 by restricting ourselves to the use of variations such that both  $\overrightarrow{AB} = \overrightarrow{CD}$  and  $\overrightarrow{AC} = \overrightarrow{BD}$  hold at the same time. This ensures that the postulates of analogy are verified for the used variations.

## 5 CBR by local enrichment of the case base

The enrichment of the case base consists in adding to the original case base  $\mathbf{CB}$  some (penalized) cases inferred from  $\mathbf{CB}$ . The inferences considered in this paper consist in applying the learned adaptation rules  $(\overrightarrow{u}, \overrightarrow{v}, c) \in \mathbf{AK}$ . In theory, all the penalized cases that can be so inferred can enrich  $\mathbf{CB}$ : this is considered in Section 5.1. However, this leads usually to a penalized case base that is too large. In Section 5.2, a local enrichment is proposed that consists in adding to  $\mathbf{CB}$  penalized cases that are “around” the target problem.

### 5.1 Theoretical view: global enrichment of $\mathbf{CB}$

The principle of global enrichment of  $\mathbf{CB}$  is simple: it consists in computing all the penalized cases  $(\mathbf{x}, \mathbf{y}, \pi)$  that can be inferred from  $\mathbf{CB}$  by application of one or several adaptation rules, with the constraint  $\pi \leq \tau_{\text{penalty}}$ , where  $\tau_{\text{penalty}}$  is a given threshold.

The size of the enriched case base,  $\mathbf{PCB}$ , can be estimated as follows. Let  $p = |\mathbf{AK}|$  and  $d = \lfloor \tau_{\text{penalty}} / \min\{c \mid (\overrightarrow{u}, \overrightarrow{v}, c) \in \mathbf{AK}\} \rfloor$ . If no hypothetical case  $(\mathbf{x}, \mathbf{y})$  is generated twice in the process, then  $|\mathbf{PCB}| = |\mathbf{CB}| \times \frac{p^{d+1} - 1}{p - 1}$ , assuming  $p \neq 1$ . Therefore,  $|\mathbf{PCB}| / |\mathbf{CB}| = \mathcal{O}(p^d)$ . For example, using  $p = 10$  adaptation rules having the same cost  $c = 1$ , if  $\tau_{\text{penalty}} = 3$ , the size of  $\mathbf{PCB}$  is about a thousand times the size of  $\mathbf{CB}$ . This illustrates the fact that this global enrichment of the case base approach produces a case base whose size is, for most CBR applications, too large, which motivates the local enrichment of the case base.



## 5.2 Practical view: local enrichment of CB

The principle of local enrichment is based first on the choice of *seed cases*, i.e., cases from CB that are chosen to produce penalized cases to be added to the case base. If  $(\mathbf{x}^s, \mathbf{y}^s)$  is a seed case, then a penalized case  $(\mathbf{x}, \mathbf{y}, \pi)$  is produced by a gradient descent starting from  $(\mathbf{x}^s, \mathbf{y}^s, 0)$ , by decreasing  $\text{dist}(\mathbf{x}, \mathbf{x}^t) + \pi$ , each step corresponding to the application of an adaptation rule. The penalized case  $(\mathbf{x}, \mathbf{y}, \pi)$  to be added to the case base thus constitutes a local optimum of the set of cases generated from the seed case  $(\mathbf{x}^s, \mathbf{y}^s)$ .

The selection of the set of seed cases SC can be done following several strategies, such as the following ones:

- The simplest strategy consists in taking all the source cases:  $\text{SC} = \text{CB}$ . This has the advantage of simplicity, but may lead to considerable growth of the case base (the enriched case base size,  $|\text{PCB}|$ , will be between  $|\text{CB}|$  and  $2|\text{CB}|$ ).
- If the size of the case base is too large already, only a few additional cases should be added and the following solutions can be proposed:
  - Choose SC by a random sampling from CB;
  - Choose SC as the  $k$  nearest neighbors of  $\mathbf{x}^t$ , e.g. according to  $\text{dist}$ .

## 6 Applications

In the remainder of the paper, we present two applications of local enrichment of the case base during case-based reasoning. Both examples create strings in a second domain (the solution space) that correspond to strings in a first domain (the problem space). The first application is a theoretical example: the problem space and the solution space are formal languages. The second application is actual machine translation: the problem space and the solution space are actual natural languages: French and English.

### 6.1 Machine translation of formal languages

The first example shows how local enrichment of the case base can be used to translate from a regular language into a context-free language. The languages we use are the prototypical examples of these families of formal languages, i.e., the problem space and the solution space are  $\mathcal{P} = \{(ab)^n \mid n \in \mathbb{N}\}$  and  $\mathcal{S} = \{A^n B^n \mid n \in \mathbb{N}\}$ , respectively. Let us suppose that our case base contains only the three smallest nonempty members of each of these languages:  $\text{CB} = \{(ab, AB), (abab, AABB), (ababab, AAABBB)\}$ .

From such a case base, in the problem space, one variation with a support greater than 1 is extracted. It corresponds to the ratio  $ab : abab = abab : ababab$ .<sup>8</sup> This variation in the problem space corresponds to a variation in the solution space:  $AB : AABB = AABB : AAABBB$ . Of course, these variations have their corresponding inverse variations in the problem and solution spaces.

<sup>8</sup> Note that this is the equality of two ratios. Of course, it is also an analogy by itself ( $ab : abab :: abab : ababab$ ), but this is not what is meant here.

| PROBLEM SPACE  | SOLUTION SPACE   | CASE BASE  |
|--|--|--|
|  |  | Initial case base<br>= Seed cases  |
|  |  | $(ab)^1$ $A^1B^1$<br>$(ab)^2$ $A^2B^2$<br>$(ab)^3$ $A^3B^3$  |
| Source problems in seed cases                          |  |  |
| $\delta = 6$ $(ab)^3$                                  |  |  |
| $\delta = 8$ $(ab)^2$                                  |  |  |
| $\delta = 10$ $(ab)^1$                                 |  |  |
|  |  | Enrichment   |
| $(ab)^1 : (ab)^2 :: (ab)^3 : \mathbf{(ab)^4}$          | $A^1B^1 : A^2B^2 :: A^3B^3 : \mathbf{A^4B^4}$          | $(\mathbf{ab})^4$ $\mathbf{A^4B^4}$  |
| $(ab)^2 : (ab)^3 :: (ab)^3 : \mathbf{(ab)^4}$          | $A^2B^2 : A^3B^3 :: A^3B^3 : \mathbf{A^4B^4}$          | $(\mathbf{ab})^4$ $\mathbf{A^4B^4}$  |
| $(ab)^1 : (ab)^2 :: (ab)^2 : (ab)^3$                   | $A^1B^1 : A^2B^2 :: A^2B^2 : A^3B^3$                   | $(ab)^3$ $A^3B^3$  |
| $(ab)^2 : (ab)^3 :: (ab)^2 : (ab)^3$                   | $A^2B^2 : A^3B^3 :: A^2B^2 : A^3B^3$                   | $(ab)^3$ $A^3B^3$  |
| $(ab)^1 : (ab)^2 :: (ab)^1 : (ab)^2$                   | $A^1B^1 : A^2B^2 :: A^1B^1 : A^2B^2$                   | $(ab)^2$ $A^2B^2$  |
| $(ab)^2 : (ab)^3 :: (ab)^1 : (ab)^2$                   | $A^2B^2 : A^3B^3 :: A^1B^1 : A^2B^2$                   | $(ab)^2$ $A^2B^2$  |
| New source problem                                     |  |  |
| $\delta = 4$ $\mathbf{(ab)^4}$                         |  |  |
| $(ab)^1 : (ab)^2 :: \mathbf{(ab)^4} : \mathbf{(ab)^5}$ | $A^1B^1 : A^2B^2 :: \mathbf{A^4B^4} : \mathbf{A^5B^5}$ | $(\mathbf{ab})^5$ $\mathbf{A^5B^5}$  |
| $(ab)^2 : (ab)^3 :: \mathbf{(ab)^4} : \mathbf{(ab)^5}$ | $A^2B^2 : A^3B^3 :: \mathbf{A^4B^4} : \mathbf{A^5B^5}$ | $(\mathbf{ab})^5$ $\mathbf{A^5B^5}$  |
| New source problem                                     |  |  |
| $\delta = 2$ $\mathbf{(ab)^5}$                         |  |  |
| $(ab)^1 : (ab)^2 :: \mathbf{(ab)^5} : \mathbf{(ab)^6}$ | $A^1B^1 : A^2B^2 :: \mathbf{A^5B^5} : \mathbf{A^6B^6}$ | $(\mathbf{ab})^6$ $\mathbf{A^6B^6}$  |
| $(ab)^2 : (ab)^3 :: \mathbf{(ab)^5} : \mathbf{(ab)^6}$ | $A^2B^2 : A^3B^3 :: \mathbf{A^5B^5} : \mathbf{A^6B^6}$ | $(\mathbf{ab})^6$ $\mathbf{A^6B^6}$  |
|  |  | Final case base  |
|  |  | $(ab)^1$ $A^1B^1$<br>$(ab)^2$ $A^2B^2$<br>$(ab)^3$ $A^3B^3$<br>$(ab)^4$ $A^4B^4$<br>$(ab)^5$ $A^5B^5$<br>$(ab)^6$ $A^6B^6$ |

**Fig. 1.** Trace for the translation from a regular language into a context-free language. The target problem is  $(ab)^6$ . It is correctly translated into  $A^6B^6$  after enrichment of the case base. The problem space is on the left, the solution space in the middle. The case base and its enrichment are shown on the right. The distance to the target problem is denoted by  $\delta$ . New source problems, solutions and cases are boldfaced; old ones are grayed out.

An actual trace of the system is given in Figure 1 for the translation of the string  $(ab)^6$ . We choose to select all cases in the case base as seed cases, i.e.,  $SC = CB$ . The seed cases are sorted by distance to the target problem. Their LCS distance to the target problem is given by  $\delta$  in Figure 1. Starting from the problems in the seed cases, applying the variation has the effect of enriching the case base with cases of the form  $((ab)^n, A^n B^n)$  from  $n = 4$  to 6, one after another. This is indeed an induction over  $n$  for  $(ab)^n$  and  $A^n B^n$  simultaneously in both spaces.

During enrichment, the distance from the new source problems to the target problem decreases down to 2. The distance of 0 is not mentioned, as it means that the new source problem is indeed the target problem, for which a solution has been found.

Such an example can be easily amended to translate from a regular language into a context-sensitive language (like  $\{A^n B^n C^n \mid n \in \mathbb{N}\}$ ), or a context-free language into a context-sensitive language. Changing the direction of translation is also possible: from context-free to regular, etc.

## 6.2 Machine translation of natural languages

The second application deals with machine translation of natural languages. We use French–English as the language pair and data from the Tatoeba Corpus<sup>9</sup> as our bilingual corpus. There are important remarks to make on this domain.

*Nature of the data:* strings of characters. This implies again that variations are defined as in Section 4.4. The case base consists of sentence pairs which are in a translation relation. We retain sentences of less than 10 words in length and select 90% of them for training and the other 10% for testing. This makes 109,390 sentence pairs in total in the training set. The average length of a sentence in French is  $6.9 \pm 1.8$  words and  $6.6 \pm 1.6$  in English. Such sentence pairs are illustrated in Figure 2. Notice that the sentences are lowercased and tokenized.

| French  | English   |
|---|---|
| <i>regardez comment je le fais !</i>                                  | <i>watch how i do it .</i>  |
| <i>vous m' avez oublié , n' est-ce pas ?</i>                          | <i>you 've forgotten me , haven 't you ?</i>                        |
| <i>pensez-vous sérieusement à vendre cela sur internet ?</i>          | <i>are you seriously thinking about selling this online ?</i>       |
| <i>les travailleurs se plaignent de leurs conditions de travail .</i> | <i>the workers are complaining about their working conditions .</i> |

**Fig. 2.** French and English example sentences, i.e., problems and solutions, in the case base for case-based machine translation

<sup>9</sup> <https://tatoeba.org/> and <http://www.manythings.org/anki/>

|   |  |  |
|---|--|--|
| <p><i>get up . : don't<br/>get up .</i></p> <p><i>tell me . : don't<br/>tell me .</i></p> <p><i>leave us . : don't<br/>leave us .</i></p> <p><i>leave me . : don't<br/>leave me .</i></p> | <p><i>he runs . : she runs .</i></p> <p><i>he is : she is<br/>young . young .</i></p> <p><i>he helps : she helps<br/>us . us .</i></p> <p><i>he loves : she loves<br/>singing . singing .</i></p> <p><i>he just : she just<br/>left . left .</i></p> | <p><i>i can't believe : i just can't be-<br/>this . lieve this .</i></p> <p><i>i need a little : i just need<br/>more time . a little more<br/>time .</i></p> <p><i>i want this to : i just want this<br/>be over . to be over .</i></p> <p><i>i want some- : i just want<br/>one to talk to . someone to<br/>talk to .</i></p> <p><i>i felt a drop of : i just felt a<br/>rain . drop of rain .</i></p> |
|---|--|--|

**Fig. 3.** Three analogical clusters which stand for variations in the solution space

*Size of the the case base.* For case-based reasoning, the case base here is quite large: 109,390 cases. However, in the field of machine translation, on the contrary, it is considered rather small.

*Nature of the ratios:* they are defined as in Section 2.2 (analogies on strings) as we deal with strings. A ratio is a vector made of the difference between the Parikh vectors of the two strings considered, plus an extra dimension with the LCS distance between the two strings. Notice again that the equality between ratios does not imply the existence of an analogy, contrary to arithmetic or geometric analogies on numbers or tuples:  $\text{dist}(A, B) = \text{dist}(C, D)$  does not imply  $\text{dist}(A, C) = \text{dist}(B, D)$ . Also, conformity is not transitive, so it is not an equivalence relation.

*Nature of the variations: analogical clusters.* Because of the nature of the ratios and the nature of conformity, variations are defined as sets of ratios in which any pair of ratios is an analogy. We use the tools<sup>10</sup> described in Fam and Lepage [5] to extract all analogical clusters containing at least 2 ratios from the case base in the problem and the solution spaces. As a note, the use a more semantically justified distance, instead of the purely formal LCS distance, is of course worth exploring: a distance associated with the cosine similarity of distributional semantics can produce semantically justified analogies (Lepage [11]) and could hence allow to produce semantic variations.

*Number of variations:* It is rather large: almost 8 million analogical clusters were extracted in French, more than half a million in English. The extraction of such variations from an actual corpus is time-consuming. For efficiency, we retain only the first 3,000 largest analogical clusters in number of ratios. Three examples of analogical clusters are given in Figure 3. Typically, variations reflect

<sup>10</sup> <https://lepage-lab.ips.waseda.ac.jp/> > Kakenhi 15K00317 > Tools

|               |   |                                      |            |
|---------------|---|--------------------------------------|------------|
| $\delta = 12$ | <i>vous êtes vraiment très bons .</i>       | <i>you are really very good .</i>    | (in CB)    |
| $\delta = 13$ | <i>vous êtes venu trop tôt .</i>            | <i>you 've come too early .</i>      | "          |
| $\delta = 14$ | <i>vous êtes venu trop tard .</i>           | <i>you came too late .</i>           | "          |
| $\delta = 14$ | <b><i>vous êtes tatillon .</i></b>          | <b><i>you 're finicky .</i></b>      | "          |
| $\delta = 14$ | <i>vous êtes venue trop tôt .</i>           | <i>you 've come too early .</i>      | "          |
| <hr/>         |   |                                      |            |
| $\delta = 4$  | <i>vous êtes vraiment très tatillon .</i>   | <i>you are really very finicky .</i> | (new case) |
| $\delta = 18$ | <i>ne vous vraiment pas êtes tatillon .</i> | <i>you really not 're finicky .</i>  | "          |
| <hr/>         |   |                                      |            |
| $\delta = 6$  | <i>vous êtes vraiment fort tatillon .</i>   | <i>you are really very finicky .</i> | "          |
| <hr/>         |   |                                      |            |
| $\delta = 0$  | <i>vous êtes vraiment trop tatillon .</i>   | <i>you are really too finicky .</i>  | "          |

**Fig. 4.** Translation process of a French sentence into English

|                         |                                       |                                    |   |
|-------------------------|---------------------------------------|------------------------------------|---|
| <i>vous êtes bons .</i> | <i>vous êtes vraiment très bons .</i> | <b><i>vous êtes tatillon .</i></b> | <i>vous êtes vraiment très tatillon .</i> |
| <i>you 're good .</i>   | <i>you are really very good .</i>     | <b><i>you 're finicky .</i></b>    | <i>you are really very finicky .</i>      |

**Fig. 5.** Variation in the problem and solution spaces resulting in the enrichment of the case base

grammatical oppositions. In the examples of Figure 3, affirmative / negative, masculine / feminine and insertion of the adverb *just*.

As an example, the translation process of the tokenized sentence *vous êtes vraiment trop tatillon .* from French into English is given in Figure 4. The seed problems are the 5 most similar French sentences in the case base. The process takes three steps corresponding to each block of sentences in Figure 4. The distance to the target problem is given on the left. Notice the infelicitous enrichment of the case base with an invalid sentence: *you really not 're finicky .*

An example of a variation applied in the problem and solution spaces during the above translation process is shown in Figure 5. It corresponds to a variation applied on the fourth seed case (boldfaced) in Figure 4. As a result, the case base is enriched with the first case marked as a new case in the above table ( $\delta = 4$ ).<sup>11</sup>

The standard metric BLEU (Papineni et al. [15]) is used for the evaluation of the accuracy of a machine translation output against a given reference set. BLEU scores range from 0 to 1; the higher, the better. The system described above achieves a BLEU score of 0.51 in translating the 10,998 sentences in the test set. Although the differences in scores are statistically significant, this is a reasonable score when compared with the scores of two much more elaborated systems, a neural system (OpenNMT<sup>12</sup>), and a statistical system (GIZA++,

<sup>11</sup> Remember that LCS distance is used:  $\text{dist}(\text{très}, \text{trop}) = 4$  (two deletions and two insertions), not 2 (two substitutions) as would be the case with Levenshtein distance.

<sup>12</sup> <http://opennmt.net>

Moses, KenLM, MERT<sup>13</sup>), which achieve 0.60 and 0.65 respectively, on exactly the same data.

## 7 Conclusion, discussion and related work

In this paper, we proposed a new approach to case-based reasoning which consists in enriching the case base while performing reasoning. Enrichment results from the application of adaptation rules to seed cases, i.e., cases taken from the case base as starting points. New cases are created, which get closer and closer to the target problem, but they get penalties characteristic of the uncertainty brought by the application of the adaptation rules. Adaptation rules are given by variations in the problem space and variations in the solution space. The last variations should approach the target problem itself, so that corresponding variations in the solution space will produce (hypothetical) solutions to the target problem. We implemented such a new approach and illustrated it with two applications which shared the fact that the solution and problem spaces were spaces of strings of characters: formal and natural languages.

The notion of penalized case is similar to the notion of *ghost case* introduced by D. Leake and B. Schack [9], a ghost case being a hypothetical case generated by adaptation. The main difference between these notions is the use of penalty values when reasoning with penalized cases. Although the aim of both works are quite opposite (case base enrichment versus case base contraction), they use these similar notions in similar ways: in [9], it is argued that ghost cases are used to compensate for the lack of expressiveness of the case base and this argument can be reused in the current work.

The general framework can be adapted to various scenarios. Several points can be adapted to the specificity of the domains at hand. For instance, the selection of the seed cases can be performed in various ways suggested in Section 5.2, at random or according to some selection method specific to the domain.

Our approach to case-based reasoning can be seen as a variant of gradient descent or hill climbing. Similarly, our approach exhibits the risk of reaching local minima (or maxima) instead of global minima (or maxima). Here, the landscape is shaped by the variations observed between the cases present in the initial case base. This issue of local optimality can be partially addressed by considering several branches generated from each seed case: instead of a single path approaching the target problem, a tree can be generated rooted at this seed case, whose breadth should be controlled to avoid an explosion of the enriched case base size. This way, several new cases can be generated from a single seed case. The precise study of this idea remains to be done.

From a theoretical viewpoint, the approach is presented in a very constrained framework, in particular for the definition of variations. This makes the explanations simpler, but, in particular for the considered applications in machine translation, some of these constraints do not hold. Thus, a theoretical study on less constrained variation spaces must be carried on.

---

<sup>13</sup> <http://www.statmt.org>

## Bibliography

- [1] Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun.* **7**(1), 39–59 (1994)
- [2] Couceiro, M., Hug, N., Prade, H., Richard, G.: Analogy-preserving functions: a way to extend Boolean samples. In: *IJCAI'17*, pp. 1575–1581 (2017)
- [3] Craw, S., Wiratunga, N., Rowe, R.C.: Learning adaptation knowledge to improve case-based reasoning. *Art. Intelligence* **170**(16-17), 1175–1192 (2006)
- [4] d'Aquin, M., Badra, F., Lafrogne, S., Lieber, J., Napoli, A., Szathmary, L.: Case Base Mining for Adaptation Knowledge Acquisition. In: *IJCAI'07*, pp. 750–755 (2007)
- [5] Fam, R., Lepage, Y.: Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages. In: *Proc. of LREC 2018*, pp. 1060–1066, ELRA (May 2018)
- [6] Hanney, K., Keane, M.: Learning adaptation rules from a case-base. In: *Advances in Case-based reasoning*, pp. 179–192, no. 1168 in *LNAI*, Springer (1996)
- [7] Hussain, Z., Gimenez, F., Yi, D., Rubin, D.: Differential data augmentation techniques for medical imaging classification tasks. In: *Annual symposium proceedings*, pp. 979–984 (2017)
- [8] Jalali, V., Leake, D., Forouzandehmehr, N.: Learning and applying adaptation rules for categorical features: An ensemble approach. *AI Comm.* **30**(3-4), 193–205 (2017)
- [9] Leake, D., Schack, B.: Exploration vs. exploitation in case-base maintenance: Leveraging competence-based deletion with ghost cases. In: *ICCBR 2018*, pp. 202–218 (2018)
- [10] Lepage, Y.: Proportional analogy in written language data. In: *Language, Production, Cognition and the Lexicon*, pp. 151–173, Springer (2014)
- [11] Lepage, Y.: Semantico-formal resolution of analogies between sentences. In: *Proc. of LTC 2019*, pp. 57–61 (2019)
- [12] Lepage, Y., Denoual, E.: Purest ever example-based machine translation: detailed presentation and assessment. *Machine Transl.* **19**, 251–282 (2005)
- [13] Lepage, Y., Lieber, J.: Case-Based Translation: First Steps from a Knowledge-Light Approach Based on Analogy to a Knowledge-Intensive One. In: *ICCBR 2018* (2018)
- [14] Lieber, J., Nauer, E., Prade, H., Richard, G.: Making the Best of Cases by Approximation, Interpolation and Extrapolation. In: *ICCBR 2018* (2018)
- [15] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. *Tech. rep.*, IBM (2001)
- [16] Richter, M.M., Weber, R.: *Case-based reasoning: a textbook*. Springer Science & Business Media (2013)
- [17] Taylor, L., Nitschke, G.: Improving deep learning using generic data augmentation. *CoRR* (2017)