Combining Multiple Similarity Metrics Using a Multicriteria Approach

Luc Lamontagne[1] and Irène Abi-Zeid[2]

[1] Department of Computer Science and Software Engineering,
Laval University, Québec, Canada, G1K 7P4
Luc.Lamontagne@ift.ulaval.ca
[2] Department of Operations and Decision Systems
Laval University, Québec, Canada, G1K 7P4
Irene.Abi-Zeid@osd.ulaval.ca

Abstract. The design of a CBR system involves the use of similarity metrics. For many applications, various functions can be adopted to compare case features and to aggregate them into a global similarity measure. Given the availability of multiple similarity metrics, the designer is hence left with two options in order to come up with a working system: Either select one similarity metric or try to combine multiple metrics in a super-metric. In this paper, we study how techniques borrowed from multicriteria decision aid can be applied to CBR for combining the results of multiple similarity metrics. The problem of multi-metrics retrieval is presented as an instance of the problem of ranking alternatives based on multiple attributes. Discrete methods such as ELECTRE II have been proposed by the multicriteria decision aid community to address such situations. We conducted our experiments for ranking cases with ELECTRE II, a procedure based on pairwise comparisons. We used textual cases and multiple metrics. Our results indicate that the use of a combination of metrics with a multicriteria decision aid method can increase retrieval precision and provide an advantage over weighted sum combinations especially when similarity is measured on scales that are different in nature.

1 Introduction

When building a CBR system, similarity metrics have to be defined in order to support case retrieval functionalities. This process involves determining a mechanism to compare the different values of each case feature (local similarity) and to aggregate these evaluations to measure the closeness of a target problem to the cases in the system's case base (global similarity). Many options at each of these steps are available and, to come up with a working CBR system, the designer must make a decision regarding the combination of metrics that will be incorporated in the retrieval component.

The motivation behind this work stems from previous results pertaining to Textual CBR [1, 2]. Lamontagne *et al.* [3] studied and compared three (3) approaches based on statistical language processing techniques for estimating the similarity of fully textual cases (*i.e.* cases where both problems and solutions are textual in nature). It was observed that the three metrics had dissimilar behaviour and that their relevance

T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 415 – 428, 2006.

varied as a function of the textual CBR systems properties such as the size of the case base, the number of neighbours in the case base, the length of the case descriptions, etc. It is therefore interesting to verify whether these metrics can be combined in order to take advantage of their individual strengths. We deem this issue worthwhile investigating for retrieval within a CBR system.

In this paper, we describe how a multicriteria aggregation approach developed within the decision aid community can contribute to combining global similarity results. Our main research question is to determine whether using multiple metrics can potentially improve performance in the retrieval phase of CBR systems. We chose the ELECTRE II method to conduct our experimentation and to verify whether it allows obtaining higher precision for case retrieval. To highlight the advantages and limitations of our multicriteria approach, we compared these experimental results with results obtained based on a weighted sum of the same metrics.

Section 2 of this paper presents information on the textual CBR background pertaining to this work and introduces the metrics used for our experimentation. In section 3, we propose a short introduction to multicriteria decision aid and present a detailed description of the ELECTRE II aggregation procedure. We explain in section 4 how the multicriteria setting is applied to CBR. We describe and discuss in section 5 our experimental results and conclude with perspectives for future work.

2 Multiple Perspectives on the Similarity of Textual Cases

The motivation behind this work stems from an investigation of textual case retrieval [3] where three (3) similarity metrics based on statistical natural language processing (NLP) methods were compared. The metrics were the following:

Cosine measure: As is frequently the case with information retrieval systems, a cosine metric (scalar product) can be applied to measure the relatedness of the cases. Case descriptions are represented as vectors with elements corresponding to individual words present in both the case problems and solutions. Words are assigned a tf*idf weight that quantifies their relative importance. For two given cases, this measures the mutual coverage of the two bags of words that define their content.

Case expansion measure: This measure relies on the expansion of case descriptions using lists of word co-occurrences. Word co-occurrences, denoting some associations between different words, are usually selected using a mutual information estimator. Case expansion is then applied on the solutions descriptions by adding words from these lists. For instance, a case containing the phrase *conference call* in its problem description could find words such as *phone number* or *dial* added to its solution. This measure tries to overcome the lexical shortcomings proper to short case descriptions by inserting additional words that might help find implicit similarities between cases.

Translation measure: This measure makes use of a statistical translation model to evaluate the probability that an existing solution was likely generated from an existing problem description. The translation model, obtained from an alignment algorithm [4, 5], computes the probability that a problem word suggests the use of

another word in the solution (this corresponds to a local similarity measure). The resulting global similarity measure is the cumulative probability that a case solution could be associated to a given target problem.

As can be seen from the preceding descriptions, these metrics are based on totally different principles. Experimental results have revealed that they had different properties and unequal performances. A cosine measure performs well for routine cases, *i.e.* problems that are frequently submitted to a system. These routine cases tend to be described using a limited number of words, which facilitates lexical comparisons. On the other hand, the two other metrics make it possible to infer associations between different words, a property that may reveal interesting for more complex case formulations. A case expansion measure is more predictive in nature but less precise for handling frequent and longer case descriptions. The translation approach has a greater potential for discriminating among word associations and is more accurate when used for providing a small number of recommended similar cases. However, this approach requires a substantial corpus in order to build a model capable of covering a large variety of problems.

Considering that the three measures can be more or less appropriate in different contexts, it is reasonable to expect an improvement in the retrieval performance of a CBR system when these three measures are combined. However, a problem may arise when the metrics are measured on very different scales. For example, in our test case base, cosine similarity values belong to a normalized scale of [0,1] and have a mean similarity value of 0.08 with a standard deviation of 0.12; whereas case expansion is measured on a non normalized scale yielding an average similarity value of 0.26 and a standard deviation of 0.11; finally the biggest challenge is to take into account the translation measure, a probability estimate of the words comprised in the case description, which very small values range from 10^{-4} to 10^{-800}. This last measure is obviously non commensurable with the cosine and case expansion measures. Although a logarithmic conversion can somehow help to exploit the translation measure, it remains difficult to combine it with the other metrics because of scale disparities.

A discrete multicriteria aggregation procedure seemed a promising direction for tackling this combination problem. This field of research has been studied for many years by the decision aid community and a multitude of techniques exist to address the problem of ranking alternatives based on multiple conflicting and non commensurable criteria. We introduce this approach in the next section.

3 Discrete Multicriteria Decision Aid

Discrete multicriteria decision Aid (MCDA) [6] provides a framework for supporting a decision maker or a group of decision makers in their decision process where a set of discrete options is considered, a set of often conflicting and non commensurable criteria is used to evaluate these options, and where the expected outcome of the process is: A recommendation of a set of good options (choice problem); a ranking of the options considered (ranking problem); or the assignment of the options considered to predefined categories (classification problem). The preferences of the decision maker(s) are modeled through a set of parameters reflecting the importance of the criteria, as well as indifference, preference and veto thresholds. A variety of aggregation

procedures exist to aggregate the local preferences (based on each criterion) into a global preference (based on all the criteria).

The decision problem is represented as a set of m options $A= (a_1, a_2, ...a_m)$, a set of n criteria $G= (g_1. g_2, ...g_n)$, and the $m \times n$ evaluations $g_j(a_i)$ of option i on criterion j expressed in a decision table **E,** i=1..m, j=1..n (Fig. 1). An interesting feature of some of the multicriteria aggregation procedure is that evaluations do not have to be on similar scales. Moreover they do not even have to be numerical in nature. For instance, one criterion can be measured on a cardinal scale with real numbers while another can be evaluated on an ordinal scale of linguistic echelons such as *weak*, *average*, *strong*.

$$\mathbf{E} = \begin{bmatrix} e_{11} = g_1(a_1) & \ldots & e_{i1} = g_1(a_i) & \ldots & e_{mn} = g_1(a_m) \\ \vdots & & \vdots & \vdots & \vdots \\ e_{1j} = g_j(a_1) & \ldots & e_{ij} = g_j(a_i) & \ldots & e_{mj} = g_j(a_m) \\ \vdots & & \vdots & \vdots & \vdots \\ e_{1n} = g_n(a_1) & \ldots & e_{in} = g_n(a_i) & \ldots. & e_{mn} = g_n(a_m) \end{bmatrix}$$

Fig. 1. An example of multicriteria decision table

Each criterion must be assigned a weight that indicates the relative importance of the criteria to the decision maker(s). The set of weights $\{\omega_j\}$ does not depend on the scales or values used for the corresponding evaluations. However, it is often assumed that the sum of weights is equal to 1.

Outranking methods are a family of multicriteria aggregation procedures based on pairwise comparisons of the options. In the following paragraphs, we describe one such popular method for ranking options, ELECTRE II.

3.1 ELECTRE II – An Multicriteria Aggregation Procedure

The multicriteria aggregation procedure we chose for our project is ELECTRE II [7, 8]. This was the first multicriteria ranking method developed based on the outranking relation principle. Given two options A and B, A *outranks* B means that A is *at least as good as* B. There are two major phases in ELECTRE II: The construction and the exploitation of the outranking relation. The construction of the outranking relation allows us to aggregate, for each pair of options, the local preferences evaluated on each criterion into a global preference structure. This means that we move from a pairwise comparison of the options based on individual criteria, to a global comparison of the pairs of options based on all the criteria. This translates into the existence or non-existence of the two following binary relations:

- $A\mathbf{P}_sB$: A strongly outranks B.
- $A\mathbf{P}_wB$: A weakly outranks B.

Once we have constructed the outranking relations between all pairs of options, we proceed to establish a direct ranking, an inverse ranking, and a final ranking. This is the exploitation phase of the outranking relation. The final ranking reflects the

decision maker(s) preferences, subject to the method, the evaluations, the preferences model, and the method's parameters.

Construction of the outranking relation. From a numerical point of view, we first need to compute concordance and discordance indices. The concordance index of a pair of options (A,B) denoted by $C(A,B)$ corresponds to the degree to which the criteria support the assertion that A is *at least as good as* B (majority rule). It is the sum of the weights of the criteria where A is evaluated equally or better than B. This is defined below where $g_j(A)$ is the evaluation of option A on criterion j and ω_j is the weight of criterion j. C is therefore a concordance matrix of $m \times m$.

$$C(A,B) = \sum_{j:g_j(A) \geq g_j(B)} \omega_j \tag{1}$$

We next compute for each criterion and each pair of options, the discordance index, $d_j(A,B)$. This denotes the degree to which criterion j does not agree with the assertion that A is *at least as good as* B. It can be interpreted as the possibility for criterion j to apply its veto (respect of minority rule) and is defined below. We must compute n discordance matrices of $m \times m$.

$$d_j(A,B) = \begin{cases} 0 & \text{if } g_j(A) \geq g_j(B) \\ g_j(B) - g_j(A) & \text{if } g_j(A) < g_j(B) \end{cases} \tag{2}$$

Once we have computed concordance and discordance indices, we apply concordance and non discordance tests in order to verify, for each pair of options (A, B) whether we have $A\boldsymbol{P}_s B$, $A\boldsymbol{P}_w B$, or no outranking relation. These tests use concordance and discordance thresholds, parameters that reflect the decision maker(s) values and preferences. There are three (3) global concordance thresholds, $0.5 < \mathbf{c_1} < \mathbf{c_2} < \mathbf{c_3} \leq 1$ and two (2) discordance or veto thresholds per criterion $0 < \mathbf{v_1}(j) < \mathbf{v_2}(j) < E(j)$, where $E(j)$ is the scale width of criterion j. When the concordance thresholds are large, we require that many criteria support the assertion that A outranks B; and for small values of the discordance (veto) thresholds, we require that none of the criteria strongly disagrees with the assertion that A outranks B. Denote by condition 1 the following:

$$\frac{\sum_{j:g_j(A) > g_j(B)} \omega_j}{\sum_{j:g_j(A) < g_j(B)} \omega_j} > 1 \qquad \textbf{condition 1} \tag{3}$$

Strong outranking test. For each pair of options (A, B): $AP_sB \Leftrightarrow$ **condition 1** is met **and**

$$\begin{aligned} &C(A,B) \geq \mathbf{c_1} \textbf{ and } d_j(A,B) \leq \mathbf{v_2}(j)\ \forall j \quad \underline{\textbf{or}} \\ &C(A,B) \geq \mathbf{c_2} \textbf{ and } d_j(A,B) \leq \mathbf{v_1}(j)\ \forall j \end{aligned} \tag{4}$$

If the pair of options (A, B) does not pass the strong outranking test, we go on to apply the weak outranking test.

Weak outranking test. For each pair of options (A, B) that do not pass the strong outranking test: $AP_wB \Leftrightarrow$ **condition 1** is met **and**

$$C(A,B) \geq \mathbf{c}_3 \textbf{ and } d_j(A,B) \leq \mathbf{v}_2(j)\ \forall j \tag{5}$$

As an illustration, consider the situation where $C(A,B)$ is high, implying that A is evaluated better than B on a set of criteria that have an important total weight, and suppose that $d_j(A,B)$ is high, meaning that B is better than A on criterion j by an important difference, larger than the veto values $\mathbf{v}_1$(j) and $\mathbf{v}_2$(j) for criterion j. This means that the data does not support the assertion that A outranks B, which does not automatically imply that B outranks A.

Exploitation of the outranking relation and construction of the final ranking. Based on the strong and weak outranking relations, we proceed to construct a direct ranking and an inverse ranking (total pre-orders). In a total pre-order, each pair of options A, B either A is ranked better than B, or B is ranked better than A, or they have the same rank. In the direct ranking, the first rank is occupied by the options that are not strongly outranked by any other options. The options in the next rank are those that are not outranked by any non-ranked options, they may be outranked by options from previous ranks, and so on. The weak outranking relation is used to differentiate between options occupying the same rank. The inverse ranking is obtained in a similar fashion. The last rank is obtained by the options that do not strongly outrank any other options. The previous rank is obtained by the options that do not outrank any non-ranked options; they may outrank options from the rank below, and so on. It is possible that some options have different ranks in the direct and inverse ranking while others end up with the same ranks.

A final ranking is obtained by combining the direct and inverse rankings through either a computation of a median rank or by intersection. In a final ranking by intersection, an option A outranks option B, if it has a higher rank in at least one of the two direct or inverse rankings, and if it has a higher or equal rank to it in the other ranking. Two options are equivalent, have the same rank, if and only if they have the same rank in both direct and inverse rankings. Two options are incomparable if and only if one has a higher rank in one of the direct or inverse rankings and a lower rank in the other ranking. Although more recent algorithms such as ELECTRE III were later developed for ranking alternatives (see [6] for specific examples), we chose ELECTRE II because it is much simpler and easier to use. Furthermore, it requires less parameters and thresholds that are somewhat arbitrary.

4 Application to CBR Retrieval

It is possible to envisage various ways to apply a multicriteria approach to CBR retrieval. Multicriteria methods could be used to aggregate local similarity values obtained at the attribute level (as in [9]). They could also be used to select the most appropriate metric as a function of case and problem characteristics. Furthermore, they can be applied, as we have done in this paper, to conduct cases retrieval based on multiple global similarity evaluations.

Given a target problem t, a case base C and a set of metrics M, we applied the ELECTRE II aggregation procedure to CBR retrieval as follows:

- Each candidate case c_i is considered an option. The decision problem consists of ranking the cases in the case base in a decreasing order of relatedness to a new problem. This leads to deciding which case(s) from the case base will be recommended as potential solutions.
- Each similarity metric m_j is a criterion of the decision process. It is assumed that the set of similarity metrics M evaluate different facets of the ability of the cases to solve a target problem t.
- The evaluations contained in the decision table correspond to the similarity measures of the target problem t with each candidate case c_i according to a specific metric m_j. Hence the evaluation $g_j(c_i) = sim_{mj}(t,c_i)$.
- The decision process consists of establishing the final ranking of the cases in the case base and of selecting the first k candidates with the highest ranks.

We present in Fig. 2 a general scheme for a multicriteria combination of the results obtained from multiple similarity metrics. *MCDADecide* is the ELECTRE II decision function described in section 3 of this paper, W is the set of weights assigned to the metrics and w_m is the relative weight of metric m.

```
MCDASelection(t, C, k, M, W) {
   // Build the decision table
   for each metric mj of M {
      for each case ci of C {
         DecisionTable[ci][mj] = sim mj (t,ci)
   }}
   // Conduct the decision process and return the first k actions
   R = MCDADecide_ElectreII(DecisionTable, W)        // a ranking of C
   S = the first k elements of R
   return S
}
```

Fig. 2. Algorithm for selecting the k most similar cases using multiple metrics and a MCDA aggregation procedure (ELECTRE II)

This strategy corresponds to a brute force application of ELECTRE II to CBR retrieval. In practice, this approach might reveal impractical for large case bases as its complexity is $O(|C|^2)$ with a significant constant. Therefore, for large scale applications, we considered two variations of this approach for limiting the number of pairwise comparisons: The bounded approach and the lexicographical approach.

In the bounded approach, the ranking results provided by the individual metrics are used to filter the cases that will be retained as options in the multicriteria aggregation process. The set of retained candidates is the union of the sets of the nearest cases based on individual metrics. The corresponding algorithm is described in Fig 3.

```
BoundedMCDASelection(t, C, k, M, W, b) {
   // Filter the candidate cases for the multicriteria process
   for each metric mj of M {
      Cj = the first b cases ci ranked according to simmj(t,ci)
      C' = C + Cj
   }
   S = MCDASelection(t, C', k, M, W)
   return S
}
```

Fig. 3. Algorithm for a bounded selection of the *k* most similar cases using multiple metrics and a MCDA aggregation procedure (ELECTRE II)

The lexicographical approach is a hierarchical approach: Cases are first filtered based on the lead metric, the one with the highest weight. Subsequently, the remaining candidate cases are ranked based on all the metrics using the multicriteria aggregation procedure. Hence the lead metric determines the candidate cases used as options while the other metrics help discriminate among them. This scheme is illustrated in Fig. 4.

```
LexicographicalMCDASelection(t, C, k, M, W, b) {
   // Filter the candidate cases to be part of the decision process
   mlead = the metric of M with the largest weight w
   C' = the first b cases ranked according to sim mlead (t,ci)
   S = MCDASelection(t, C', k, M, W)
   return S
}
```

Fig. 4. Algorithm using a lead metric to filter candidate cases before selecting the *k* most similar ones

5 Experimental Analysis

Tests were conducted using 73 cases from an Email Response application, where a case consists of a request message (the problem) and its corresponding response (the solution). Since these cases are textual in nature, we used the three metrics described in Section 2 of this paper to measure similarity. The results presented in this section were obtained from a leave-one-out evaluation of the retrieval component. In order to evaluate the performance of various combinations of metrics, we assessed the best k cases according to the following indicators:

- *Precision*: The proportion of relevant cases in the first *k* nearest-neighbours (*k*=5, for this experiment); Cases are considered relevant when case solutions share common themes, which indicates that a response can be reused.
- *Relevant First*: The proportion of trials for which the nearest neighbour is relevant.

5.1 Individual vs. Combined Metrics

The first issue was to determine whether a multicriteria combination of metrics can provide a better performance than using one metric at a time. As presented in Table 1,

experimental results indicate that a combination of the three metrics can improve the performance of the system by approximately 5 to 7 % (in terms of precision and relevance of the first recommendation). These results were obtained by assigning a weight distribution of W = {0.25, 0.5, 0.25} to the Cosine, Case Expansion and Translation measures respectively. We observed throughout our experimentations that similar results could be obtained if higher weights were assigned to the Case Expansion measure. However, this improvement in performance was not significant when higher weights were assigned to the Cosine and Translation measures, in which case the precision obtained was 0.6.

Table 1. Performance using individual and a combination of similarity metrics with ELECTRE II.

Similarity metric	Precision	Relevant First
Cosine measure	0.57	0.58
Case Expansion measure	0.61	0.68
Translation measure	0.56	0.63
Multicriteria combination of the three metrics	0.64	0.73

To better understand the influence of each metric on system performance, we used combinations of pairs of metrics with equal weights of 50%. The results, presented in Table 2, indicate that all the MCDA pairs outperformed the precision of their constituents when used individually. One intriguing observation is that the combination of Cosine and Case Expansion measures provides the same performance as a MCDA combination of the three metrics.

Table 2. Performance of MCDA combination of pairs of similarity metrics

Similarity metric	Precision	Relevant First
Cosine + Case Expansion	0.64	0.73
Cosine + Translation	0.60	0.62
Case Expansion + Translation	0.63	0.62

Fig. 5 clearly shows that when the majority of weight is assigned to the Case Expansion metric, then using the other metric helps increase the global precision of the MCDA retrieval combination. Otherwise, the precision either remains constant or degrades. For instance, in Fig. 5a, we observe some improvement of performance when weight values inferior to 0.5 are allocated to the Cosine metric. An abrupt decrease in precision occurs when more weight is assigned to this metric. Therefore, we can draw the conclusion that improvement can be expected from MCDA combinations of metrics when the best performing metric (Case expansion) has a higher weight coefficient.

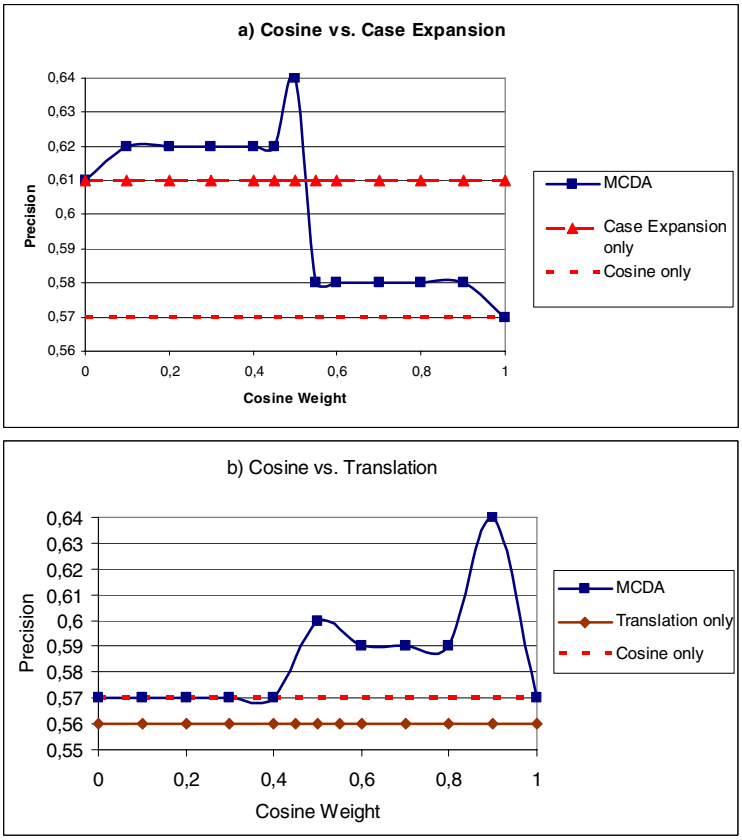

Fig. 5. Effect of weight variation on the performance of MCDA pairs of metrics

5.2 Bounding the Number of Cases Before Applying the MCDA Combination

The results we obtained are presented in Fig. 6. The bounded version of MCDA combinations (algorithm described in Fig. 3) has a slight degradation of performance of approximately 1.5% in precision when the number of cases used in the aggregation procedure is between 5 and 10. However, when more than 13 cases are used as options, it offers a precision either equal or higher than a brute force MCDA combination. Also, the relevance of the first case (not shown on this figure) is on average as good as the Brute Force MCDA approach when the case limit is above 5.

The lexicographical version is less stable and presents a slower performance improvement than the preceding approach. We note that a higher case limit (> 20) is required to reach performances equal or superior to the basic MCDA combination. On the other hand, our experiments indicate that the relevance of the first case is not influenced when using more than 6 cases.

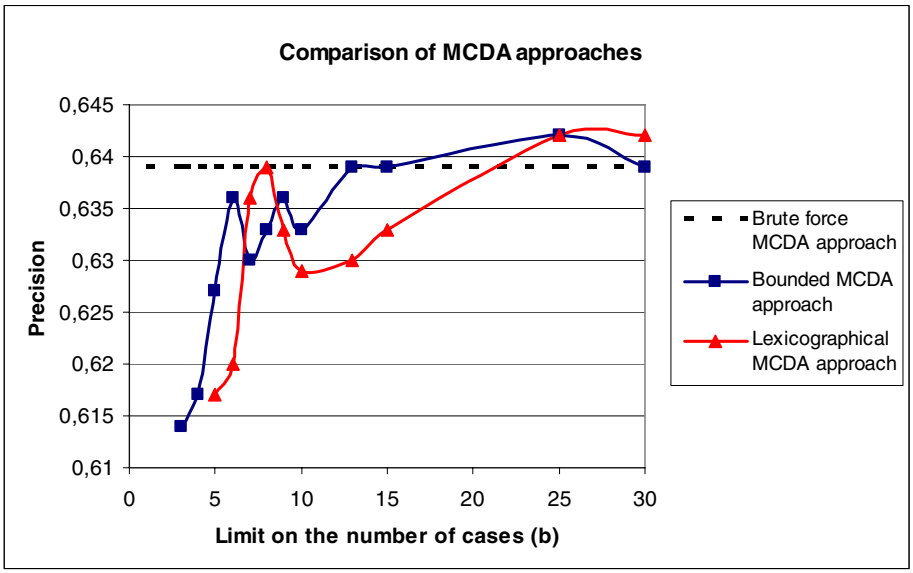

Fig. 6. Effect of limiting the number of cases on the performance of MCDA approaches

5.3 MCDA Combination vs. Weighted Sum Combination

A question may arise regarding the pertinence of using MCDA combinations as opposed to a weighted sum of the metrics. To help answer this question, we present in Table 3 comparison results where the same set of weights is used in order to ensure comparisons on the same basis.

Table 3. Comparison of the performance of a MCDA combination of the three metrics and the weighted sum (W = {0.2, 0.5, 0.3})

Similarity metric	Precision	Relevant First
MCDA combination	0.64	0.73
Weighted Sum	0.57	0.62

This table seems to indicate that the MCDA combination is a better choice than the weighted sum. However, if we perform an ablation study of the metrics (Fig. 7), we observe the following: Fig. 7a) shows that, for Cosine and Case Expansion metrics, MCDA and Weighted Sum combinations behaved similarly.

This is explained by the fact that the scales for both metrics are similar. The Weighted Sum can provide higher precision if a weight assignment for the Cosine metric is carefully chosen (weight interval ranging from 0.3 to 0.45); however it performance is degraded when the Cosine weight is in the 0.1-0.3 interval. Fig. 7b) presents a different picture. The weighted sum combination fails to outperform the other approach for the large majority of the weight assignments. Moreover, almost no

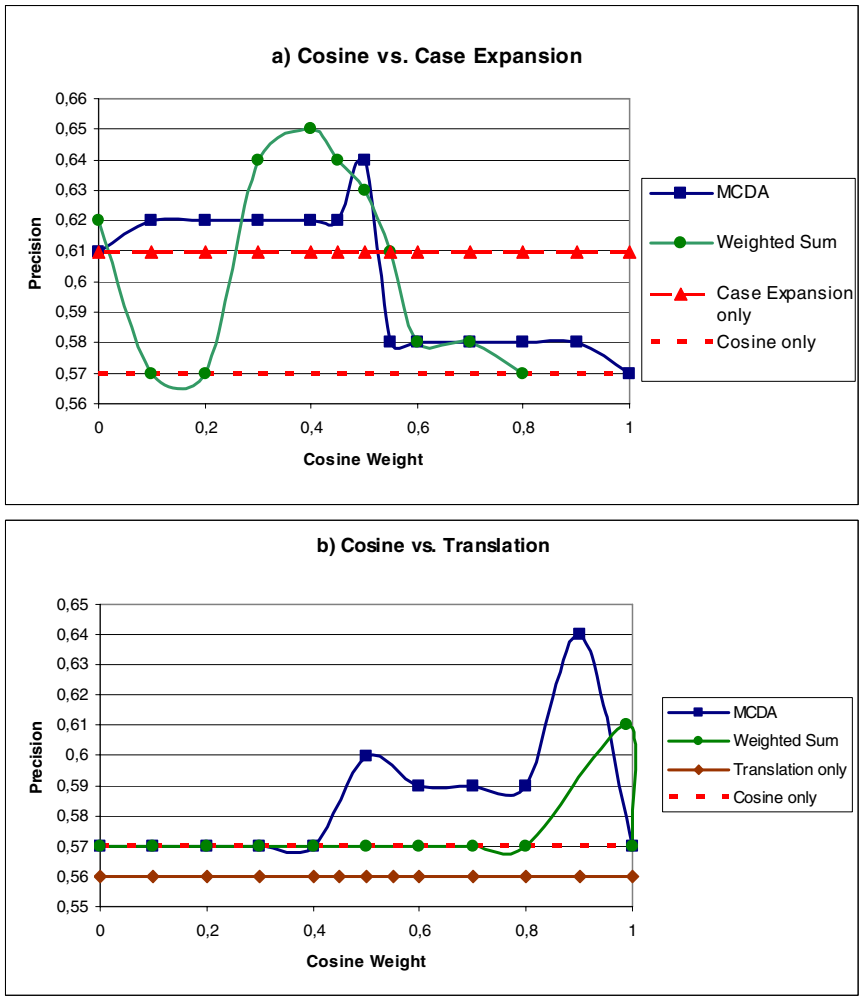

Fig. 7. Effect of weight variation on MCDA and Weighted sum combinations

improvement can be attained except when weight of the Cosine metric in the 0.8-0.9 interval. Therefore, the scale difference between the two Cosine and Translation metrics seems to affect significantly the performance of the weighted sum approach.

This behaviour can be explained by the difference in the magnitudes of the metrics and the weights. In the weighted sum approach, the weights are used to establish a compromise between the scales of the various metrics, hence making both components strongly dependent on each other. The weighted sum is a completely compensatory method where a small evaluation on one metric is cancelled by a high evaluation on another metric. In the MCDA approach, evaluations are used for pairwise comparisons of actions with respect to a single metric. Evaluations from different

metrics are not explicitly aggregated. Furthermore, the weights are used for the concordance and discordance computations. Therefore, the MCDA approach is not dependent on the closeness of the weights magnitudes and the metrics scales.

6 Conclusion

In this paper, we have explored how a discrete multicriteria aggregation procedure can be used to combine metrics for case ranking in a case retrieval process. The motivation behind this work was to investigate whether CBR systems could benefit from using multiple metrics simultaneously.

Our results indicate that multicriteria combinations can improve the performance of individual metrics. This approach revealed particularly advantageous when metrics are evaluated on different non commensurable scales. One interesting finding is that the output quality of the multicriteria procedure depends on the relative importance of the most performing metrics.

In order to reduce the computational burden, we proposed filtering strategies that allowed reducing the number of cases used as options in the MCDA ranking process, without sacrificing performance.

As future work, the engineering of multi-metrics CBR systems will require techniques, based on machine learning, to assist the designer in the assignment of relative weights to criteria in the ranking process. Sensitivity analysis will also help evaluate the impact on the ranking process of the various parameters used by the aggregation method. Experiments with aggregation procedures other than ELECTRE II, will help understand if the outranking approach is viable for CBR retrieval. We foresee even greater potential for applications where the similarity computations exploit non numerical syntactic and semantics properties of the cases. Multi-metric combinations can then take into account various perspectives for evaluating textual similarity. Finally, MCDA pairwise comparisons of cases should be investigated to assist other phases of the CBR cycle such as maintenance and case authoring.

References

1. Lamontagne, L. ; Lapalme, G. ; (2002) "Raisonnement à base de cas textuels – état de l'art et perspectives", Revue d'Intelligence Artificielle, Hermes, Paris, vol. 16, no. 3, pp. 339-366.
2. Lenz, M.; Hübner, A.; Kunze, M.; (1998) "Textual CBR", in Lenz, M.; Bartsch-Spörl, B.; Burkhard, H.-D.; Wess, S. (Editors), *Case-Based Reasoning Technology: From Foundations to Applications*, Lecture Notes in Computer Science 1400, Springer, pp. 115-138.
3. Lamontagne, L. ; Langlais, P. ; Lapalme, G. ; (2003) "Using Statistical Models for the Retrieval of Fully-Textual Cases", in Russell, I.; Haller, S. (Editors), Proceedings of FLAIRS '03, AAAI Press, Ste-Augustine, Florida, pp.124-128.
4. Knight, K. (1999), A Statistical Machine Translation (MT) Tutorial Workbook, unpublished document, *http://www.isi.edu/natural-language/mt/wkbk.rtf*.
5. Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., Roossin, P., (1990), "A Statistical Approach to Machine Translation", *Computational Linguistics*, 16(2), pp. 79-85.

6. Vincke, Ph.; (1992) *Multicriteria decision aid*, J. Wiley, New York.
7. Roy, B. and Bertier, P. (1973), La méthode ELECTRE II : une application au média-planning, *Proceedings of the sixth IFORS'72*, Ross (ed.), North-Holland Pub.
8. Simpson, L, (1996), "Do decision makers know what they prefer?: MAVT and ELECTRE II", *Journal of the Operational Research Society*, 47(7): pp. 919-929.
9. San Pedro, J., Burstein, F. (2003) "A Framework for Case-Based Fuzzy Multicriteria Decision Support for Tropical Cyclone Forecasting", Proceedings of HICSS 2003, ACM Press, pp. 85-92.