

# Qualitative Evaluation of Automatically Calculated Acception Based MLDB

Areë Teeraparseree

GETA, CLIPS, IMAG

385, rue de la Bibliothèque

B.P. 53 - 38041 Grenoble Cedex 9, France

aree.teeraparseree@imag.fr

## Abstract

In the context of the Papillon project, which aims at creating a multilingual lexical database (MLDB), we have developed Jeminie, an adaptable system that helps automatically building interlingual lexical databases from existing lexical resources. In this article, we present a taxonomy of criteria for evaluating a MLDB, that motivates the need for arbitrary compositions of criteria to evaluate a whole MLDB. A quality measurement method is proposed, that is adaptable to different contexts and available lexical resources.

## 1 Introduction

The Papillon project<sup>1</sup> aims at creating a cooperative, free, permanent, web-oriented environment for the development and the consultation of a multilingual lexical database. The macrostructure of Papillon is a set of monolingual dictionaries (one for each language) of word senses, called *lexies*, linked through a central set of interlingual links, called *axies*. *Axies*, also called *interlingual acceptions*, are not concepts, but simply interlingual links between lexies, motivated by translations found in existing dictionaries or proposed by the contributors. Figure 1 represents an interlingual database that links monolingual resources in three languages: French, English and Japanese. The interlingual acceptions (*axies*) are linked to lexies from each language. For instance, a lexie for the French word “terre” is linked through an *axie* to two lexies for the English words “earth” and “soil” and to a lexie for the Japanese word “tsuchi”. Note that an *axie* can be refined into a set of axes. For instance, a lexie for the English word “chair” is linked through *axie1* to two lexies for the French words “fauteuil” and “chaise”. *Axie1* can be refined into two axes *axie11* and *axie12* as illustrated in figure 2.

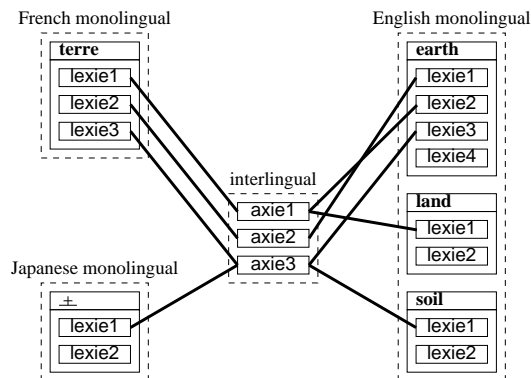


Figure 1: An example interlingual database

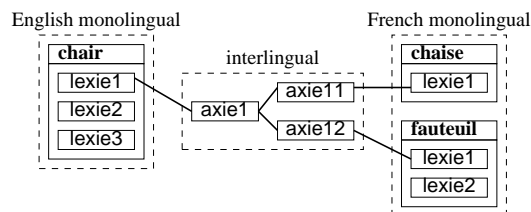


Figure 2: An example of refined axes

This pivot macrostructure has been defined by (Sérasset, 1994) and experimented by (Blanc, 1999) in the PARAX mockup. The microstructure of the monolingual dictionaries is the “DiCo” structure, which is a simplification of Mel’cuk’s (Mel’cuk et al., 1995) DEC (Explanatory-Combinatorial Dictionary) designed by Polguère & Mel’cuk (Polguère, 2000) to make it possible to construct large, detailed and principled dictionaries in tractable time.

The building method of the Papillon lexical database is based on one hand on 1) reusing existing lexical resources, and on the other hand on 2) contributions of volunteers working through Internet. In order to automate the first step, we have developed Jeminie (cf. section 2), a

<sup>1</sup><http://www.papillon-dictionary.org/>

flexible software system that helps create (semi-) automatically interlingual lexical databases. As there are several possible techniques for the creation of axes that can be implemented in Jeminie, it is necessary to evaluate and compare these techniques to understand their strengths and weaknesses and to identify possible improvements. This article proposes an approach for the automatic qualitative evaluation of an automatically created MLDB, for instance created by Jeminie, that relies on an evaluation software system that adapts to the measured MLDB.

The next section of this article provides an overview of the Jeminie system and the strategy it implements to create interlingual lexical databases. The third section presents in detail evaluation criteria for an MLDB. The fourth section describes the evaluation system that we propose and the metrics and criteria to evaluate the quality of MLDB. Last sections discuss the measurement strategy and conclude.

## 2 Jeminie

Jeminie is a software system that helps building interlingual databases. Its first function is to automatically extract information from existing monolingual dictionaries, at least one for each considered language, and to *normalize* it into lexies. The second function of Jeminie is to automatically link lexies that have the same sense into axes. The prominent feature of Jeminie is the ability to arbitrarily *combine several axie creation techniques* (Teeraparbsee, 2003).

An axie creation technique is an *algorithm* that creates axes to link a set of existing lexies. An algorithm may use existing additional lexical resources, such as: bilingual dictionaries, parallel corpora, synonym dictionaries, and antonym dictionaries. Algorithms that do not rely on additional lexical resources consider only information available from the monolingual databases, and include vectorial algorithms such as calculating and comparing conceptual vectors for each lexie (Lafourcade, 2002).

The use of one algorithm alone is not sufficient, in practice, to produce a good quality MLDB. For instance, using only one algorithm that uses bilingual dictionaries, one obtains a lexical database on the level of words but not on the level of senses of words. The Jeminie system tackles this problem from a *software engineering* point of view. In Jeminie, an axie creation al-

gorithm is implemented in a *reusable software module*. Jeminie allows for arbitrary composition of modules, in order to take advantage of each axie creation algorithm, and to create a MLDB of the best possible quality. We call a MLDB *production process*, a sequence of executions of axie creation modules. A process is specified using a specific language that provides high-level abstractions. The Jeminie architecture is divided into three layers. The *core layer* is a library that is used to implement axie creation modules at the *module layer*. The *processes interpreter* starts the execution of modules according to processes specified by linguists. The interpreter is developed using the core layer. Jeminie has been developed in Java following object-oriented design techniques and patterns.

Each execution of an axie creation module progressively contributes to create and filter the intermediate set of axes. The final MLDB is obtained after the last module execution in a process. The quality of a MLDB can be evaluated either 1) on the final set of axes after a whole process has been executed, or 2) on an intermediate set of axes after a module has been executed in a process. The modularity in MLDB creation provided by Jeminie therefore allows for a wide range of quality evaluation strategies. The next sections describe the evaluation criteria that we consider for MLDBs created using Jeminie.

## 3 Taxonomy of evaluation criteria

Here, we propose metrics for the qualitative evaluation of multilingual lexical databases, and give an interpretation for these measures. We propose a classification of MLDB evaluation criteria into four classes, according to their nature.

### 3.1 Golden-standard-based criteria

In the domain of machine translation systems, an increasingly accepted way to measure the quality of a system is to compare the outputs it produces with a set of reference translations, considered as an approximation of a golden standard (Papineni et al., 2002; hovy et al., 2002). By analogy, one can define a golden standard multilingual lexical database to compare to a database generated by a system such as Jeminie, that both contain axes that link to lexies in the same monolingual databases. Considering that two axes are the same if they contain links to exactly the same lexies, the quality of a machine generated multilingual lexical database

would then be measured with two metrics adapted from machine translation system evaluation (Ahrenberg et al., 2000): recall and precision.

*Recall (coverage)* is the number of axes that are defined in both the generated database and in the golden standard database, divided by the number of axes in the golden standard.

*Precision* is the number of axes that are defined in both the generated database and in the golden standard database, divided by the number of axes in the generated database.

However, (Aimelet et al., 1999) highlighted the limits of the golden standard approach, as it is often difficult to manually produce precise reference resources. In the context of the Papillon project, a golden standard multilingual lexical database would deal with nine languages (English, French, German, Japanese, Lao, Thai, Malay, Vietnamese and Chinese), which makes it extremely difficult to produce. Furthermore, since the produced multilingual lexical database in Papillon will define at least 40000 axes, using heterogeneous resources, a comparison with a typical golden standard of only 100 axes seems not relevant. Instead of producing a golden standard for a whole multilingual lexical database, we propose to consider *partial golden standard* that concerns only a part of a MLDB. For instance, a partial golden standard can be produced using a bilingual dictionary that concerns only two languages in the database. Several partial golden standard MLDBs could be produced using several bilingual dictionaries, in order to cover all languages in the multilingual lexical database.

### 3.2 Structural criteria

Structural evaluation criteria consider the state of links between lexies and axes. We define several general structural criteria:

- $CLA_{ave}$ , the average number of axes linked to each lexie. Here, we consider only lexies that are linked to axes.  $CLA_{ave}$  should be 1. If it is  $> 1$ , several axes have the same sense, i.e. the produced MLDB is ambiguous. If it is  $< 1$ , the produced MLDB may not be precise enough, as it does not cover all the lexies. Actually, we should also consider the standard deviation of that number, because a MLDB would be quite bad if  $CLA_{ave} = 2$  for half the lexies and  $CLA_{ave} = 0$  for the rest, although the global value of  $CLA_{ave}$  is 1.

- for each language,  $ADL_{lang}$ , the ratio of the number of axes to the number of lexies in that language. If it is too low, the axes may represent fuzzy acceptions. If it is too high, axes may overlap, i.e. several axes may represent the same acceptance. Typically, it should be about 1.2 (cf. large MLDB such as EDR - the Electronic Dictionary Research project in Japan). This metrics should be calculated for each language independently, because the number of lexies may significantly vary between two languages, making this metrics irrelevant if calculated using the total number of lexies and axes in a database.
- $CAL_{ave}$ , the average number of lexies of each language linked to each axis. It should be about 1.2. If it is  $> 1$  for a language, axes may represent a fuzzy acceptance or there is synonymy, as illustrated in figure 3. If it is  $< 1$  for a language, axes may not cover that language precisely. Note that  $CAL_{ave}$  may help us locate places in the “axis” set where an axis is refined by one or more axes. Each  $CAL_{ave}$  may then be far from  $CAL_{ave}$  global, but their average should still be near  $CAL_{ave}$  global for the considered set.

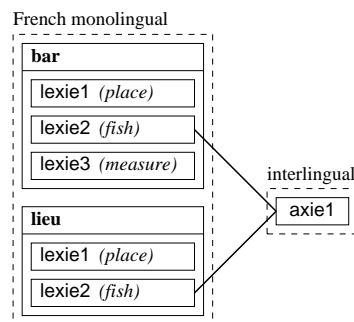


Figure 3: Example of two lexies that are synonym in the same language and linked to the same axis

Such metrics are complementary and can easily be measured, and are among the rare metrics that concern a whole MLDB. They, however, do not help evaluating the quality of links between axes and lexies in terms of semantics.

### 3.3 Human-based criteria

This class of evaluation criteria is based on the measurement of the number and nature of the *corrections made by a linguist* on a part of a

produced MLDB. For instance, one can measure the ratio of the number of corrections made by a linguist, to the total number of links between the considered axes and lexies. The closer the ratio is to zero, the higher is the quality of the multilingual lexical database. A high correction ratio implies a low MLDB quality.

However, this class of criteria assumes that the produced MLDB are homogeneous. In the context of Papillon, the database will be produced using several techniques and heterogeneous lexical resources, which limits the relevance of such criteria.

This approach is similar to the golden-standard approach described above, although the golden-standard approach is automatic.

### 3.4 Non-resource-based semantic criteria

In this class, criteria evaluate the quality of the semantics of the links between axes and lexies, and do not rely on additional lexical resources. One of the metrics that we consider is the distance between conceptual vectors of lexies linked to the same axis. A conceptual vector for a lexie is calculated by projecting the concepts associated with this lexie into a vector space, where each dimension corresponds to a leaf concept of a thesaurus (Lafourcade, 2002). The concepts associated with a lexie are identified by analyzing the lexie definition. The lower the distance between the conceptual vectors of two lexies is, the closer are those lexies (word-senses). As a metrics, we therefore consider the average conceptual distance between each pair of lexies linked to the same axis. The lower that value is, the better the MLDB is, in terms of the semantics of the links between axes and lexies. However, a reliable computation of conceptual vectors relies on the availability precise and rich definitions in lexies, and on large lexical resources to compute initial vectors, which are difficult to gather for all languages in practice.

### 3.5 Discussion

As a more general conceptual framework, we define a classification of evaluation criteria along four *dimensions*, or characteristics:

- *automation*: a criterion is either automatically evaluated, or relies on linguists.
- *scope*: a criterion evaluates either a part of a MLDB, or a whole MLDB.

- *semantics*: a criterion considers either the structure of a MLDB, or the semantics of the links between axes and lexies.
- *resource*: a criterion relies on additional lexical resources, or not.

Multilingual lexical databases such as Papillon can be used in different contexts, e.g. in machine translation systems or in multilingual information retrieval systems. The criteria used for evaluating a multilingual lexical database should be *adapted to the context* in which the database is used. For instance, if a multilingual lexical database is very precise and good at French and Japanese acceptions, but not good at other languages, it should be judged as a good lexical database by users who evaluate a usage of French and Japanese only, but it should be judged as a bad multilingual lexical database globally.

Since the Papillon database generated by Jemie will not be tied to specific usages, the database production system must not impose predefined evaluation criteria. We propose instead to allow for the use of any criterion at any point in the four dimensions above and for *arbitrary composition of evaluation criteria* to adapt to different contexts. However, since we aim at performing an automatic evaluation, we do not consider human-based criteria, although human evaluation is certainly valid. Our approach is similar to the approach chosen in Jemie for the creation of axes. We tackle this problem of criteria composition from a software engineering point of view, by using object oriented programming techniques to design and implement modular and reusable *criterion software modules*.

## 4 Adaptable evaluation system

By analogy with the Jemie modules that implement algorithms to create axes, we propose a system that allows for the implementation in Java of reusable software modules that implement algorithms to measure MLDB. In this system, we consider that each criterion is implemented as a module. Criterion modules are of a different kind, and are developed differently from Jemie axis creation modules. As a convention, we define that each criterion module returns a numeric value as the result of a measurement, noted  $Q_i$ . The higher that value, the better the evaluated database.

#### 4.1 Axie-creation-related criteria

As the strategy we have chosen in Jeminie is to *combine complementary axie creation modules* to produce axes in a multilingual lexical database, we consider that each axie creation module encapsulates its own quality criterion that it tends to optimize, explicitly or implicitly. Since each module implements an algorithm to *decide whether to create an axie*, we consider that such an algorithm can also be used as a criterion to *decide whether an existing axie is correct*. An axie creation module can not be reused as is as a criterion module, however its decision algorithm can be easily reimplemented in a criterion module. For each algorithm, we define the following four metrics, adapted from (Bédécarrax, 1989):

- $A_1$  the number of *internal adjustments*, i.e. the number of axes that would be created according to the algorithm, and that have actually been created.
- $A_2$  the number of *external adjustments*, i.e. the number of axes that would *not* be created according to the algorithm, and that have actually *not* been created.
- $E_1$  the number of *internal errors*, i.e. the number of axes that would *not* be created according to the algorithm, and that have actually been created.
- $E_2$  the number of *external errors*, i.e. the number of axes that would be created according to the algorithm, and that have actually *not* been created.

For each algorithm, the quality criteria are to maximize  $A_1 + A_2$ , to minimize  $E_1 + E_2$ , or to maximize  $(A_1 + A_2) - (E_1 + E_2)$ .

#### Resource-based algorithms

For instance, following are the definitions of  $A_1$ ,  $A_2$ ,  $E_1$  and  $E_2$  for the axie creation algorithm that uses a bilingual dictionary between languages X and Y:

- $A_1$  the number of pairs of lexies of languages X and Y that are linked to the same axie and which words are mutual translations according to the bilingual dictionary.
- $A_2$  the number of pairs of lexies of languages X and Y that are not linked to the same axie and which words are not mutual translations according to the bilingual dictionary.

$E_1$  the number of pairs of lexies of languages X and Y that are linked to the same axie and which words are not mutual translations according to the bilingual dictionary.

$E_2$  the number of pairs of lexies of languages X and Y that are not linked to the same axie and which words are mutual translations according to the bilingual dictionary.

However, resources used by resource-based creation algorithms have a number of entries that is often significantly lower than the number of lexies and axes in a multilingual lexical database. For instance, the number of translation entries in a bilingual dictionary is typically lower than the number of available monolingual acceptations in the source language, because that set of lexies may be constructed by combining a set of rich monolingual dictionaries. For instance, our monolingual database for French contains about 21000 headwords and 45000 lexies extracted from many definition dictionaries such as Hachette, Larousse, etc. Our monolingual database for English contains about 50000 headwords and 90000 lexies extracted from English WordNet 1.7.1. However, the bilingual French-English dictionary that we use is based on the FeM<sup>2</sup> multilingual dictionary, and defines only 15000 French headwords.

lexical database	number of headwords
French monolingual	21000
English monolingual	50000
FeM	15000

Table 1: Comparing the number of entries in monolingual lexical databases with the number of entries in the multilingual lexical database

According to the example above, measuring the number of external adjustments  $A_2$  and internal errors  $E_1$  is therefore not relevant. For example, a criterion can not decide if the words of a French lexie and of an English lexie that are linked together, are translations of each other, since the bilingual dictionary used is not precise enough. We therefore propose a *simplified quality criterion* for resource-based algorithms, that is to maximize  $A_1$  and to minimize  $E_2$ .

<sup>2</sup>French-English-Malay dictionary <http://www-clips.imag.fr/geta/services/fem>

## Vectorial algorithms

This measure can also be adapted to the comparison of the conceptual distance between lexies:

$A_1$  the number of pairs of lexies that are linked to the same axie and which conceptual vector distance is below a given threshold.

$A_2$  the number of pairs of lexies that are not linked to the same axie and which conceptual vector distance is above the threshold.

$E_1$  the number of pairs of lexies that are linked to the same axie and which conceptual vector distance is above the threshold.

$E_2$  the number of pairs of lexies that are not linked to the same axie and which conceptual vector distance is below the threshold.

This algorithm is not limited by the size of an additional lexical resource, and can decide whether any pair of lexies should be linked or not. It is therefore possible to evaluate  $A_2$  and  $E_1$  in addition to  $A_1$  and  $E_2$ .

## Synthesis

We specify that the value returned by such axie-creation-related criteria is calculated as  $Q_i = A_1 - E_2$  for resource-based criteria, and as  $Q_i = (A_1 + A_2) - (E_1 + E_2)$  for any other axie-creation-related criteria, as those formulas reflect both the number of adjustments and the number of errors.

### 4.2 Structural criteria

As described above, structural criteria consider the structure of each axie in a whole multilingual lexical database. We propose to implement such algorithms also as modules in our system. For example, we define one criterion module to calculate the following value:

$$Q_i = \frac{1}{0.01 + \left| 1 - \frac{\sum_{k=1}^{nblexies} \frac{nblinkedaxies_k}{nblexies}}{nblexies} \right|}$$

where  $nblexies$  is the total number of lexies in the database, and  $nblinkedaxies_k$  is the number of axes linked to a lexie  $k$ .  $Q_i$  is comprised between 0 and 100.

### 4.3 Global criteria

A global quality value  $Q$  can be calculated as the sum of each quality value measured by each

measurement module. The choice of the measurement modules corresponds to a given usage context of the evaluated database, and the positive weight of each metric module in this context is specified as a factor in the sum:

$$Q = \sum_{i=1}^{nbmodules} weight_i \cdot Q_i$$

The objective is to maximize  $Q$ . The weight for each module can be chosen to emphasize the importance of selected criteria in the context of evaluation. For instance, when specifically evaluating the quality of axes between French and English lexies, the weight for a bilingual EN-FR dictionary-based criterion module could be higher than the weights for the other criterion modules. In addition, the values returned by different criterion modules are not normalized. It is therefore necessary to adapt the weights to compensate the difference of scale between  $Q_i$  values.

## 5 Evaluation method

One can evaluate the quality of a MLDB after it has been created or enhanced through the execution of an axie creation process by Jeminie. Such a quality measure can be used by linguists to *decide* whether to execute another axie creation process to enhance the quality of the database, or to stop if the database has reached the desired quality. The creation of an axie database is therefore *iterative*, alternating executions of axie creation processes, quality evaluations, and decisions.

It should be noted that the execution of an axie creation process may not always imply a monotonous increase of the measured quality. Since axie creation algorithms may not be mutually coherent, the order of executions of modules, in a process or in several consecutively executed processes, has an impact on the measured global quality. More precisely, the additional resources used by axie creation modules, and/or by quality criteria modules, may contain errors and be mutually incoherent. The execution of a resource-based axie creation module using a resource  $R_1$ , can cause a drop of the  $A_1$  value and an increase of the  $E_2$  value measured by a resource-based criterion module using a resource  $R_2$  incoherent with  $R_1$ . This may significantly decrease the evaluated global quality. The database may however be actually of a better quality if  $R_2$  has a poor quality and  $R_1$  has a

good quality. This highlights the need for good quality resources for both creating the database and evaluating its quality.

Another problem is that the additional lexical resources used, such as bilingual dictionaries, generally provide information at the level of words, not at the level of senses. It is thus necessary to complement these resource-based axie creation modules, for instance by using vectorial modules. Moreover, it is necessary to develop new algorithms to increase the internal consistency of an axie database, for example one that merges all the axes that link to the same lexie.

## 6 Example processes

Figure 4 illustrates the two sets of axes created by a process A and a process B to link to lexies retrieved from a French and an English monolingual dictionaries. Process A consists of the execution of only module Mbidict, that uses a bilingual dictionary FR-EN extracted from FeM dictionary and partially illustrated in figure 5. The set of axes produced by process A consists of axie1 to axie7. Process B consists of the execution of the same module Mbidict as in process A, then of a module Mvect that implements a conceptual vector comparison algorithm for filtering some bad links. Process B produces only axie1, axie4, axie5 and axie7. Note that processes A and B were hand-simulated in this example.

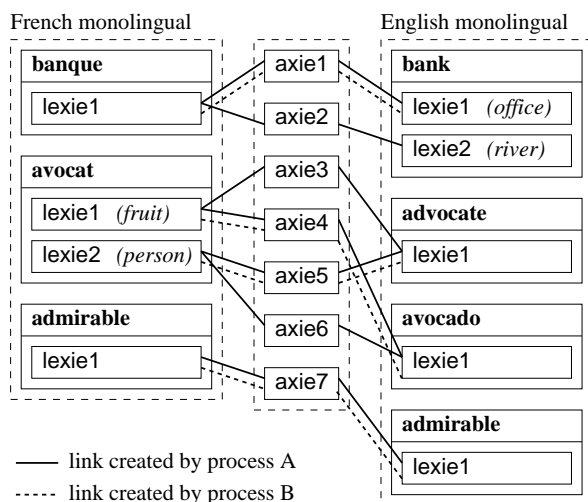


Figure 4: Axes created by processes A and B

The two same criterion modules are used to evaluate both processes: 1) an axie-creation-related criterion module using the same bilingual dictionary as the one used in the axie creation modules in processes, and calculating

Bilingual Dictionary FR-EN (FeM)

Banque (n.f.)	—	Bank (n.)
Admirable (a.)	—	Admirable (a.)
Avocat (n.m.)	—	Advocate (n.)
Avocat (n.m.)	—	Avocado (n.)

Bilingual Dictionary EN-FR (Le Robert & Collins)

Bank (n.)	—	Banque (n.f.)
Bank (n.)	—	Rive (n.f.)
Admirable (a.)	—	Admirable (a.)
Advocate (n.)	—	Avocat (n.m.)
Avocado (n.)	—	Avocat (n.m.)

Figure 5: Bilingual dictionaries

a  $Q_{bidict}$  value, and 2) the structural criterion module described in section 4.2, and calculating a  $Q_{struct}$  value. The global evaluated quality value for the set of axes created by each process is:

$$Q = \alpha \cdot Q_{bidict} + \beta \cdot Q_{struct}$$

The actually evaluated values of  $Q_{bidict}$  and  $Q_{struct}$ , and of  $Q$  for several combinations of  $\alpha$  and  $\beta$ , are shown in table 2.

	process A	process B
$Q_{bidict}$	7	1
$Q_{struct}$	1.76	8.25
$Q (\alpha=1, \beta=1)$	8.76	9.25
$Q (\alpha=1, \beta=2)$	10.52	17.5
$Q (\alpha=2, \beta=1)$	15.76	10.25

Table 2: The results of qualitative evaluations

Axie creation module Mbidict considers only words, but not senses of words. It therefore creates several axes linked to each lexie, some of which are not correct because they do not distinguish between the lexies of a given translation word. In process B, module Mvect is executed to suppress links and axes that are semantically incorrect. The structural quality, as given in  $Q_{struct}$ , is therefore better with process B than with process A, and intuitively the global quality has actually increased. However, executing module Mvect reduces the quality from the point of view of a bilingual translation that considers only words and not acceptions, as given in  $Q_{bidict}$ .

This illustrates that not all quality criteria should be maximized to attain the best possible quality. Weight factors for each criterion module should be carefully chosen, according to the scale of the values returned by each module, and to the linguistic objectives. For instance, as illustrated in table 2, setting a weight too high for the bilingual translation criterion lets the evaluated global quality decrease, while it has actually increased.

## 7 Conclusion

This article presents the problem of the automatic creation and evaluation of interlingual multilingual lexical databases (MLDB), in the context of the Papillon project. It describes the Jeminie software system, that we are developing, for the automatic creation of interlingual acceptions (axies). It can adapt to different contexts, e.g. to different lexical resources and different languages, by providing a means to arbitrarily compose axie creation modules.

We have proposed a taxonomy of criteria for the automatic evaluation of a MLDB. One criteria alone is not sufficient to significantly evaluate the quality of a whole database. We therefore propose a method for the arbitrary composition of evaluation criteria, following the same principles as the Jeminie system.

The proposed method will be implemented in a software framework, along with a library of modules that implement a variety of evaluation criteria, and that can be freely composed. This framework will be integrated with Jeminie, in order to allow for the automatic evaluation of a MLDB during its creation.

## References

- Lars Ahrenberg, Magnus Merkel, Anna Sagvall Hein, and Jorg Tiedemann. 2000. Evaluation of word alignment systems. In *Proceeding of LREC'2000*, pages 1255–1261, Athens, Greece.
- Elisabeth Aimelet, Veronika Lux, Corinne Jean, and Frédérique Segond. 1999. WSD evaluation and the looking-glass. In *Proceedings of TALN'1999*, Cargèse, France.
- Chantal Bédécarrax. 1989. *Classification automatique en analyse relationnelle : la quadri-décomposition et ses applications*. thesis, Université Paris 6.
- Etienne Blanc. 1999. PARAX-UNL: A large scale hypertextual multilingual lexical database. In *Proceedings of 5th Natural Language Processing Pacific Rim Symposium*, pages 507–510, Beijing. Tsinghua University Press.
- Eduard hovy, Margaret King, and Andrei Popescu-Belis. 2002. Principles of context-based machine translation evaluation. *Machine Translation*, 17(1):43–75.
- Mathieu Lafourcade. 2002. Automatically populating acception lexical databases through bilingual dictionaries and conceptual vectors. In *Papillon'2002 Seminar*, Tokyo, Japan.
- Igor Mel'cuk, André Clas, and Alain Polguère. 1995. *Introduction à la lexicologie explicative et combinatoire*. Duculot, Louvain-la-Neuve.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceeding of ACL'2002*, pages 311–318, Philadelphia.
- Alain Polguère. 2000. Towards a theoretically motivated general public dictionary of semantic derivations and collocations for French. In *Proceedings of EURALEX'2000*, pages 517–527, Stuttgart.
- Gilles Sérasset. 1994. Interlingual lexical organisation for multilingual lexical database in NADIA. In *Proceedings of COLING'94*, volume 1/2, pages 278–282, Kyoto, Japan.
- Aree Teeraparbserree. 2003. Jeminie: A flexible system for the automatic creation of interlingual database. In *Papillon'2003 Seminar*, Sapporo, Japan.