Metrics for Translation Quality Assessment: A Case for Standardising Error Typologies

Arle Lommel

Abstract Translation quality assessment (TQA) has suffered from a lack of standard methods. Starting in 2012, the Multidimensional Quality Metrics (MQM) and Dynamic Quality Framework (DQF) projects independently began to address the need for such shared methods. In 2014 these approaches were integrated, centring on a shared error typology (the "DQF/MQM Error Typology") that brought them together. This approach to quality evaluation provides a common vocabulary to describe and categorise translation errors and to create translation quality metrics that tie translation quality to specifications. This approach is currently (as of 2018) in the standardisation process at ASTM International and has seen significant uptake in industry, research, and academia. By bringing together disparate strands of quality assessment into a unified systematic framework, it offers a way to escape the inconsistency and subjectivity that have so far characterised TQA.

Keywords Translation quality assessment · Principles to practice · DQF · MQM · Translation errors · Translation metrics · Translation specifications · Standardisation

1 Introduction

This article provides an overview of three systems for translation quality assessment (TQA): (i) the Multidimensional Quality Metrics (MQM) framework,[1] developed by the author and colleagues at the German Research Centre for Artificial Intelligence (DFKI) in Berlin, Germany from 2012 through 2014 as part of the European Union-funded QTLaunchPad project; (ii) the TAUS Dynamic Quality Framework

[1] http://qt21.eu/mqm-definition/

A. Lommel (✉)
Common Sense Advisory (CSA Research), Indiana University, Bloomington, IN, USA

J. Moorkens et al. (eds.), *Translation Quality Assessment*, Machine Translation: Technologies and Applications 1, https://doi.org/10.1007/978-3-319-91241-7_6

(DQF) Error Typology,[2] developed by the Amsterdam-based Translation Automation User Society (TAUS); and (iii) the harmonisation of the two, carried out as a collaborative effort by DFKI and TAUS within the EU-funded QT21 project in 2014 and 2015.

These projects had the common goal of improving the current state-of-the-art for TQA and to address the lack of best practice approaches in this area. Although MQM and DQF began separately and many perceived them as competing projects, they were successfully harmonised to create an emerging de facto standard for TQA, one that is now in the formal standardisation process at ASTM.

This chapter first provides a history of TQA in the translation and localisation industry, which developed along lines separate from those used in Translation Studies or machine translation (MT) research. It then describes the MQM framework in some detail before turning to the DQF Error Typology. It ends with a description of the harmonised error typology and closes with a description of plans for the future.[3]

2 Historical Background

As the translation industry emerged from a cottage industry in the late 1980s and shifted towards a technology-driven one serving global enterprises in the 1990s, it became evident that its ad hoc and subjective quality evaluation methods left much to be desired. At the time, best practice emphasised having a bilingual reviewer or a monolingual subject matter expert review translations and give an assessment. Such assessments were typically informal, in the sense that they did not use formal, predefined rubrics or tools for evaluation. As a result, actual practice varied from one translation provider to another and one reviewer to another, and this inconsistency was also a source of confusion for clients.

As an example of how subjective such review could be, consider the case of one US-based language service provider (LSP) that provided multiple languages for a large client that manufactured computer peripherals. In the mid-1990s it systematically sent its translations to third-parties for review. In one instance, the company had received a Korean-language translation of a manual from a trusted and reliable translator and sent it on to another linguist. This individual sent it back with a scathing review, in which he stated that the translation was unusable and would need to be completely redone from scratch.

After the company received this review, it took the unusual step of sending the translation to a second reviewer and asked him for his opinion and to confirm

[2] https://www.taus.net/knowledgebase/index.php?title=Error_typology

[3] Because most of this chapter is written from the perspective of the author, who was active in development of MQM and the MQM/DQF harmonisation effort – and who had previously led development of the LISA QA Model – much of the account contained here does not cite published sources. For details of MQM and DQF, please see the relevant online resources cited herein.

the judgement of the first reviewer. After examining the translation, the second reviewer explained that Korean has seven formality levels and that the translator had chosen one that was moderately formal, but the initial reviewer had wanted a more formal level. Due to how extensively Korean marks these levels at the grammatical level, changing levels would have required a complete rewrite of the translation. Fortunately for this LSP, the second review agreed with the translator and said that it could go as it was and that it was an excellent translation.

Unfortunately, such disagreements were a common occurrence and buyers of translation had little guidance in how to interpret such disagreements. Contributing to the problem was that feedback was often quite vague, consisting of a subjective impression couched in imprecise terms – "the style is off," "the translation is clumsy," "it needs to be reworked," etc. – that gave the translator little, if any, concrete feedback. As a result, many translators saw quality review steps as a way for unscrupulous clients to penalise them or renegotiate prices. At the same time, LSPs faced pressure from their clients to assure them that translations were adequate, and sometimes saw translator resistance to these methods as an evasion of responsibility.

2.1 Early Efforts Toward Systematic Quality Evaluation

One way that LSPs tried to improve the situation was by using translation "scorecards" for projects. These tools, typically in spreadsheet form, allowed reviewers to count numbers of errors to generate overall quality scores, usually represented as a percentage, with 100% indicating no errors. They usually included anywhere from 2 to 15 categories of errors. In some cases, reviewers simply counted errors for each category, but in others, they also assigned them weights – such as minor, major, and severe – that incurred different penalties.

Although these spreadsheets created the impression of objectivity, their categorisations remained ad hoc and varied from LSP to LSP. They also did not tie their counts to specific locations in the text. These characterisations resulted in two problems: (i) the scores were ultimately unverifiable because the only link to the text was in the mind of the reviewer; (ii) it was unclear if the scores they generated correlated with audience or customer requirements.

The first of these problems could be addressed in part if reviewers took copious notes or marked-up hard copies of the text, but these approaches were time-consuming and introduced manual steps into the review process. They did not create audit trails to ensure that required changes were made. For example, if a note said something like "awkward style, p.2, paragraph 3, second sentence," it would provide guidance for a reviser, but there was no way to mark the location digitally or confirm that appropriate steps were taken to address the issue.

The second problem was more severe. Ad hoc categorisations might work well in some cases, but not in others. Their application across heterogeneous content types resulted in an assessment method suitable for one text type being used for others

with very different requirements. For example, if a scorecard that emphasised style were used for internal documentation aimed at service technicians, the translator might be unfairly penalised for failing to meet a certain level of style that was irrelevant to its audience. In addition, a text might receive exemplary marks on one scorecard and negative marks on another, calling into question the objective nature of the scores.

2.2 The Beginnings of Standardisation

The 1990s witnessed two systematic efforts to address the ad hoc nature of TQA:

- **SAE J2450**. This standard, from SAE International, developed a simple scorecard-style metric for automotive documentation. It featured six error types and two severity levels. GM reported that using it on a 5% sample of its texts enabled the company to simultaneously improve quality, time to market, and cost by substantial margins (Sirena 2004).
- **LISA QA Model**. Although never formally standardised, the LISA QA Model served as a de facto standard for quality assessment of software and documentation localisations after its release in the 1990s. It was developed within the auspices of the now-defunct Localization Industry Standards Association (LISA), which released it as a spreadsheet and later as stand-alone software. The Model featured 18 or 21 categories (there was a disagreement between its user interface and documentation) and three severity levels. It allowed customisation for two content types (documentation and software user interface) and included some issues specific to localisation into East Asian languages. A 2015 study found that the LISA QA Model remains the most common shared model for quality assessment in the translation industry, 4 years after its last commercial availability. However, most users modified it to some extent to meet their needs (Snow 2015).

Both efforts moved in the direction of shared metrics and promoted transparency across projects and translation providers. In theory, either of these allowed translation requesters to compare scores over time and to obtain them from various sources.

In practice, these approaches faced important limitations. First, low inter-annotator agreement (IAA) meant that reviewers were not interchangeable, particularly with regard to how severe they felt errors to be: Two reviewers, faced with the same error, might disagree as to how important it was, or even if was an error. Second, the effort to standardise error types only added to the problem that models that had been developed with specific scenarios or text types in mind might not be appropriate for different scenarios or text types. The frequency with which the LISA QA Model was modified shows that users felt the need to adapt it to their circumstances.

SAE J2450 explicitly addressed the problem of scope by stating up front that it was intended only for automotive service manuals and that other content types

would require their own metrics. However, implementers were often not so careful or principled in their application of the model, and used it for other content types, including marketing materials. The LISA QA Model was developed with software documentation and UIs in mind, but its larger inventory of error types inadvertently encouraged the notion that it was a comprehensive translation quality metric.

By the late mid-2000s, the limitations of these models were increasingly evident and discussion began in LISA about a successor to the LISA QA Model that would address them. This effort resulted in internal documents and prototypes for a standard, tentatively called Globalization Metrics Exchange – Quality (GMX-Q) that was intended to be a companion to the existing LISA standard GMX-V (for the volume of translation) and the planned GMX-C (for representing the complexity of a source text). However, the closure of LISA in 2011 prevented the release of GMX-Q and only internal working drafts existed at that time.

Subsequent to the demise of LISA, two groups began active work on translation quality assessment: (i) the Translation Automation User Society (TAUS) developed its Dynamic Quality Framework (DQF), a collection of multiple approaches to the subject; (ii) the EU-funded QTLaunchPad project, led by the German Research Centre for Artificial intelligence (DFKI), took up the work carried out in GMX-V and developed an extensive translation error typology for use in detailed analysis of human and machine translation.[4]

The following sections provide an overview of MQM and the DQF error typologies and then I move on to describe the harmonised error typology based on them.

3 Overview of the Multidimensional Quality Metrics (MQM)

This section focuses on the MQM error typology. It covers the basics of the MQM and describes the approach and principles it adopts. As discussed above, MQM took up work and ideas previously developed by LISA's GMX-Q effort. In particular, it adopted the following principles:

- **A flexible catalogue of error types**. Rather than creating a list of issues that apply to all translation and content types, the MQM developers created a master vocabulary for describing translation errors. It is not intended that the list of types be applied in its entirety. Instead, adopters select issue types relevant to their needs and apply them. This principle means that MQM does not define a single metric, but rather a common vocabulary for declaring metrics. In this respect, it closely resembles the TermBase eXchange (TBX, ISO 30042:2008) standard, which provides an XML vocabulary to describe terminology databases.

[4]See Popović in this volume for a discussion of the application of MQM to MT.

- **Compatibility with existing specifications and tools**. Rather than attempting to create a categorisation from scratch, MQM examined existing specifications and tools to capture their approaches and harmonise them. The goal was to provide an easy path for tools to adopt MQM without changing their functionality more than necessary.
- **A hierarchical approach**. Not all assessment activities require the same degree of detail. For example, the project in which MQM was created detailed categories for types of grammatical errors, but most production evaluations would require only a single overall category for them. As a result, MQM has a tree-like structure in which categories have child types that can be used for greater specificity.
- **A specifications-based approach**. To address the problem of metrics that did not tie to requirements, and to be fair to translation providers, MQM strongly emphasises the use of documented translation specifications. Based on ASTM F2575 (ASTM 2014), specifications detail what is expected of the parties involved in translation production. Any assessment method should check only things that were actual requirements. For example, if specifications state that style is not important, reviewers should not penalise translations for problems with style.

3.1 MQM Harmonised Existing Approaches

Work on MQM started with a detailed comparison of nine existing error typologies and tools: the LISA QA Model, SAE J2450 and ISO CD 14080[5]; the error categories from SDL Trados Classic, ApSIC XBench, Okapi CheckMate, the XLIFF:doc specification, and Yamagata QA Distiller; and the grading rubric for ATA certification examinations. The number of issue types contained in these varied from 6 (SAE J2450) to 23 (Okapi CheckMate). In addition, the comparison considered the documentation of the original LISA QA Model, which enumerated sub-types of its basic list of issues, and raised the total to 65. The project harmonised the names of these issue types into a superset with 145 items (Table 1).

Of these 145 error types, only one was found in all of the examined metrics and tools (adherence to terminological guidelines) and only 23 were found in more than one. The most common error types were: Terminology/glossary adherence (9), Omission (8), Punctuation (6), Consistency (5), Grammar/syntax (5), Spelling (5), Mistranslation (4), and Style (4).

The MQM typology developed a superset of the tools it examined, with two exceptions, i.e. it omitted issue types related to project satisfaction from the LISA QA Model (1) and extremely detailed issues (2):

[5]This committee draft in ISO TC37 was subsequently withdrawn and bears no relationship to the current ISO 14080, a standard for management of greenhouse gases.

Tool name	Type	Number of issues
ApSIC XBench	Automatic checker	9
ATA Certification Exam	Grading exam for human translators	21
ISO CD 14080	Proposed standard for quality assessment	21
LISA QA Model	Quasi-standard and proprietary scorecard	21 (UI) 18 (doc. Top level) 65 (full doc.)
Okapi CheckMate	Automatic checker	23
SAE J2450	Standard	6
SDL TMS Classic	Human assessment tool in Trados suite	7
XLIFF:doc	Module in translation tool exchange format	11
Yamagata QA Distiller	Automatic checker	20
TOTAL		**145**

Table 1 Issue types in existing error typologies and tools

1. The LISA QA Model contained issue types related to overall project performance, such as adherence to deadlines, completeness of deliverables, compatibility of delivered software with external applications, and functional problems with localised software. Although these issues are very important, they were outside the scope of linguistic QA and were better addressed within the scope of ISO standards such as the ISO 9000 series or ISO 17100:2015. Accordingly, these were moved into a deprecated "compatibility" branch in MQM, with a caution that they should not be used.
2. In some instances, the tools examined had very fine-grained distinctions. For example, they had a combined total of 12 issues related to white-space within translation segments. In such cases, MQM adopted a more parsimonious approach and declared a single category, such as "whitespace," rather than try to capture all possible detail.

This effort and subsequent feedback within the QTLaunchPad project resulted in initial drafts of MQM that contained 104 issue types.[6] This number subsequently increased to 182 in the final version,[7] largely due to the inclusion of additional specific types of errors related to internationalisation that were not initially present. Here it must be re-emphasised that the creators of MQM never intended that any application would use all, or even most, of these categories. Instead, they were included to provide a way to systematically describe more task-appropriate metrics.

[6]See http://www.qt21.eu/mqm-definition/mqm-spec-2014-02-14.html#hierarchical_list for a full list.

[7]See http://www.qt21.eu/mqm-definition/issues-list-2015-05-27.html

3.2 *Overall Structure of MQM*

3.2.1 Hierarchy

MQM is highly hierarchical. The hierarchy of issue types extends up to four layers of increasing specificity, although most of the hierarchy stops at the second or third layer. Figure 1 illustrates this hierarchy: *Design* is a first-level issue type (also called a "dimension," as described in the next section), *Local formatting* a second-level issue type, *Font* a third-level, and *Bold/italic* is at the fourth level. Each layer of the hierarchy provides more specific instances of the parent. For example, *Bold/italic* is also an example of *Font*, *Local formatting*, and *Design*, and can be categorised at any of these levels, depending on whether a specific metric includes them or not.

It is important to note that children of an issue type are not comprehensive: they are not intended to enumerate all possible cases of the parent. For example, the issue type *Font* has three children: *Bold/italic*, *Single/double-width* (a reference to the width of character in East Asian Languages), and *Wrong size*. These encapsulate common issues with fonts, but if a reviewer found that a serif font had been used where a sans serif font was called for, there is no specific subtype for this and *Font* would be used.

As a general principle, MQM metrics should be as course-grained as possible while still serving their purpose. Fine-grained categories are often difficult to

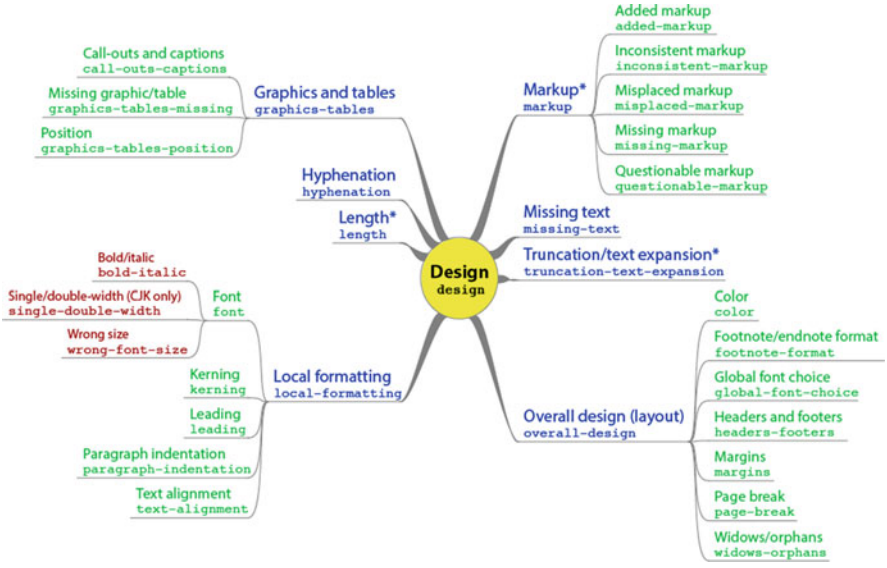

Fig. 1 Hierarchical structure in MQM. (Source: http://www.qt21.eu/mqm-definition/issues-list.html. Note that this graphic includes two names for each issue. The top name is a human-readable name, which might be translated. The second is an ID for the issue, which remains constant and can serve as a valid identifier in XML and in most programming languages)

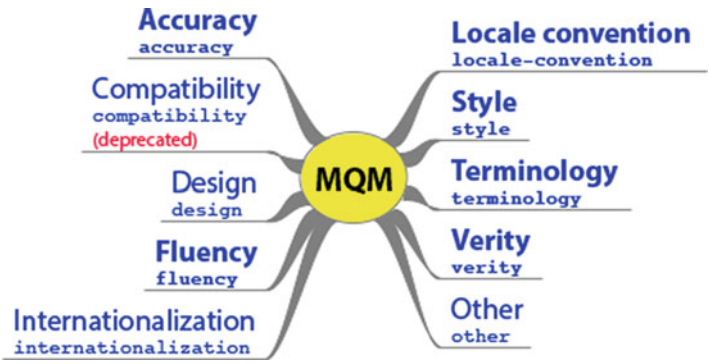

Fig. 2 Top-level "branches" or "dimensions" of MQM. (Taken from http://www.qt21.eu/mqm-definition/issues-list-2015-05-27.html)

distinguish: testing in the QTLaunchPad project showed that annotators could often agree on a high-level category while remaining uncertain about lower-level ones. In cases of uncertainty, parent categories should be used because they represent the point of least uncertainty and their children also count as examples of their parents.

3.2.2 Dimensions

At the top level in the MQM hierarchy are eight primary "branches" or "dimensions" (Fig. 2).[8] They contain other issues.[9]

These branches are defined as follows:

- **Accuracy (18 issues)**. Issues related to the relationship of the meanings conveyed by the source and target content. It applies to cases where the propositional content of the source is incorrectly rendered in the target.
- **Design (33 issues)**. Issues related to the physical appearance of the content, e.g. formatting, desk-top publishing.
- **Fluency (39 issues)**. Issues related to the linguistic well-formedness of the content. These apply to *all* texts, regardless of whether they are translations or not.
- **Internationalisation (49 issues)**. Issues related to how well the content is prepared for localisation. These issues usually are detected through problems with the target content but indicate an engineering or design fault in the source.

[8]This total does not include the "Compatibility" branch, which is used only to represent project-related issues from the LISA QA Model, and "Other," which is used for anything that does not fit into other branches.

[9]The full hierarchy and list of issues is available at http://www.qt21.eu/mqm-definition/issues-list.html

- **Locale convention (14 issues)**. These issues address whether correctly translated content is displayed correctly for the target locale. For example, dates are displayed in different formats depending on the locale, and using an incorrect one can lead to confusion.
- **Style (7 issues)**. Issues related to the overall feel of a text or adherence to style guides.
- **Terminology (7 issues)**. Issues related to the use of domain-specific terminology in the content. These are separated from general accuracy or fluency because localisation processes typically manage terminology separately.
- **Verity (7 issues)**. Issues dealing with the relationship of the content to the world in which it exists.

Each of these branches can serve as an issue type on its own if no further specificity is needed. For example, an MQM metric that consisted of two issue types – e.g. *Accuracy* and *Fluency* – would be considered valid.

The last of the branches, *Verity*, is a novel contribution from MQM to the broader study of translation quality. It addresses cases in which a translation may be accurate and fluent and yet is not appropriate for the "world" or environment in which it is used. For example, consider a US English text about electrical systems that describes ground wires as *bare copper*. If this text were to be accurately translated as *bare Kupfer* into German, it would be problematic in Germany because ground wires there are covered in green and yellow-striped insulation. This problem is not directly with the translation itself, but rather in the relation between the text and the environment in which it will be used: If the translation were intended for German speakers in the U.S., *bare Kupfer* would be a correct and appropriate rendering of the text.

Verity errors frequently occur in legal or regulatory texts, which often require substantial adaptation to correctly refer to laws and regulations in the environment in which they occur. They also commonly occur even in technical documentation in cases where source texts may refer to call centers or service options that differ between locales. For example, if a text translated from German into Chinese contains only a phone number for a German support call center, even though a Chinese support number exists, it will convey inappropriate information to users.

3.2.3 Specifications

One difficulty in standardising translation quality assessment has been that most efforts are universalising. In other words, they try to set forth a set of criteria that *all* translations *should* follow, regardless of purpose or other requirements. By contrast, a *functionalist* or *Skopos*-oriented theory of translation emphasises that translations serve purposes and should be evaluated in terms of them (Nord 1997; see also Sect. 2.1 in Castilho et al., this volume). Without a knowledge of the intended purpose, an evaluator cannot say whether the translation fulfils it. This intended purpose is not always obvious from the source text because the target-language purpose may be dramatically different from that of the source-language. For example, a

government body might intercept communication between two would-be terrorists who are trying to encourage each other. The translation of such a text should not aspire to be persuasive, but instead to convey nuance and detail relevant to the needs of an intelligence analyst who has to decide what action to undertake to prevent an attack.

MQM embraces a functionalist perspective at its core. For an MQM-based metric to be valid, it must measure how well a translation meets specifications. Accordingly, MQM metrics should be tied to a set of relevant specifications that follow ASTM F2575-14, which defines a standard set of 21 "parameters" that describe the information needed to complete a translation project (Fig. 3).

A. Linguistic work product parameters [1–13]

A.1. Source-content information [1–5]

1. textual characteristics
    a. source-language
    b. text type
    c. audience
    d. purpose
2. specialized language
    a. subject field
    b. terminology
3. volume
4. complexity
5. origin

A.2. Target content requirements [6–13]

6. target-language requirements
    a. target-language
    b. target-terminology
7. audience
8. purpose
9. content correspondence
10. register
11. format
12. style
    a. style guide
    b. style relevance
13. layout

B. Process tasks [14–15]

14. typical tasks
    a. preparation
    b. initial translation
    c. in-process quality assurance
15. additional tasks

C. Project Environment [16–18] and Relationships [19–21]

16. technology
17. reference materials
18. workplace requirements
19. permissions
    a. copyright
    b. recognition
    c. restrictions
20. submissions
    a. qualifications
    b. deliverables
    c. delivery method
    d. delivery deadline(s)
21. expectations
    a. compensation
    b. communication

Fig. 3 Translation project parameters defined in ASTM F2575:2014

To create an MQM metric, one determines which issue types are needed to check compliance with the specifications. During evaluation, if problems arise that the chosen issue types do not address, this typically indicates that the metric does not follow the specifications or that the specifications themselves were deficient and need to be revised.

Tying metrics to specifications helps ensure that evaluators do not try to fix issues they see that do not need to be fixed, or even make fixes that do the translation harm. This also enables requestors to tie feedback to requirements that translation providers had in advance, rather than providing feedback that seems to change the nature of the job after the fact.

Note that there is no expectation in MQM that new specifications and metrics will be created for every project. Such a practice would be wasteful and confusing. Instead, the intention is that implementers create specification templates and default metrics for different project types and reuse them to promote clarity and consistency.

3.2.4 Severities and Weights

When evaluating a translation, it is typically not enough to know how many errors are present. Evaluators also need to know (a) how severe they are and (b) how important the error type is for the task at hand. Severity and importance are distinct concepts in MQM.

Severity refers to the nature of the error itself and its effects on usability of the translation. The more severe an error is, the more likely it is to negatively affect the user in some fashion. Severity applies to individual errors, not to categories as a whole.

By default, MQM supports four severity levels:

1. **Critical**. Critical errors are those that by themselves render a project unfit for purpose. Even a single critical error would prevent a translation from fulfilling its purpose (e.g. by preventing the intended user from completing a task) and may have safety or legal implications. For example, if a translation of a text describing weight limits for an industrial centrifuge converts “2 pounds” into “2 kilograms” (instead of “0.9 kilograms”), it could result in destruction of the equipment or injury of its user, and is a critical error. These errors are especially problematic if the issue is not obvious in the translation.
2. **Major**. Major errors make the intended meaning of the text unclear in such a way that the intended user cannot recover the meaning from the text, but are unlikely to cause harm. They must be fixed before release, but if they were not they would not result in negative outcomes (other than possibly annoyance for the user). For example, if a translation of an educational book about insects renders the Italian *ape* (‘bee’) as *monkey* in English because *ape* is a false friend, the intended meaning may not be recoverable from the text, but it is unlikely to result in negative outcomes.
3. **Minor**. Minor errors are those that do not impact usability. In most cases the intended user will correct them and move on, perhaps without even noticing

them. For example, if an English translation says "to who it may concern" instead of "to whom it may concern," no meaning is lost and many, perhaps a majority, of readers will not even notice the slight grammatical error. Because they do not affect usability, minor errors do not need to be fixed prior to distribution.

4. **Null**. The null level is used to mark changes that are not errors. For example, if the requestor decides to change a term after a translation is submitted, the reviewer could mark *Terminology* issues with this severity. No penalties are applied at this severity. It was added to MQM in 2014 to improve compatibility with the TAUS DQF.

Each severity level corresponds to penalty points that are used in scoring translations. The default penalties are 100 points (critical), 10 points (major), 1 point (minor), and 0 points (null). Previous metrics had tended to assign values much closer to each other (the LISA QA Model used values of 1, 5, and 10), but consultation with experts in evaluation indicated that these did not provide a distinction in value sufficient to guide evaluators. For those who wish to emulate older systems, the values assigned to each level can be adjusted, although doing so impedes comparison with scores generated using the default values.

By contrast, *importance* refers to the relative value assigned to different categories of errors, rather than to individual instances. For example, someone could say that style is not important for their technical documentation, meaning that even tremendously awkward style would not matter very much if the intended meaning comes across. On the other hand, such problems might be very important for their marketing materials, because those are selling a brand image where style is crucial. Importance is addressed through the use of *weights*.

Implementers assign weights to particular issue types. They indicate how important particular issues are and allow metrics to adjust how much they contribute to overall scores. The default weight in MQM is 1.0. Higher numbers indicate greater importance and lower ones indicating that issues are not as important. For example, if a content creator determines that terminology compliance is particularly important, it could assign a weight of 1.5 to *Terminology*, which would mean that all errors related to Terminology count 50% more than the default value. By contrast, if the creator determines that *Style* is not particularly important, it could assign a weight of 0.5, which indicates a 50% reduction in any penalties assigned to it.

Although weights had been a concept in earlier quality metrics (such as the LISA QA Model), they had generally not been implemented in software. As a result, few implementers currently use weights, but they do provide a mechanism to reflect relevant priorities within evaluation.

3.2.5 Scoring

One can use an MQM metric to evaluate a translation. If scoring is desired (rather than just identification of errors), the MQM definition suggests a default scoring model. To calculate a score, one takes each error, multiplies it by its severity value and its weight to generate penalty points. If default weights are used, a minor

error is thus 1 penalty point, a major is 10, and a critical is 100. These points are then summed up to obtain the total. The score is then calculated per the following formula:

$$Score = 1 - \frac{Penalties}{WordCount}$$

The resulting score is typically presented as a percentage. As an example, if a translation has 500 words and the reviewer finds 3 minor errors and 2 major errors (23 penalty points), the score would be 95.4%. Note that negative scores are possible with this model if the penalty points exceed the number of words.

If it is desirable, scores can also be calculated for any issue type or branch within MQM by summing up the points for any issues it contains and applying the same formula. This ability allows implementers to understand what issue types contribute to quality problems and take remedial action. For example, if grammar and spelling errors pose a particular problem, an LSP or translation requester might add a requirement that the translation pass a grammar and spelling check before submission. On the other hand, if the translation receives low marks for terminology-related problems, that would suggest that the translators should receive training in terminology and apply tools to check and enforce proper terminology usage.

If implementers use scores to determine acceptance, they also need to set thresholds for what constitutes an acceptable translation. Thresholds should be set by applying the metric to translations the requesters found acceptable and some that they were unsatisfied with to find the score below which problems are evident. This value will vary between adopters of MQM and there is no universal threshold.

3.2.6 Holistic vs. Analytic Evaluation

Within translation evaluation, two general approaches apply: *holistic* and *analytic*. Holistic evaluation looks at the translation as a whole and attempts to evaluate its quality based on overall criteria. By contrast, analytic evaluation considers and analyses individual errors.

These two methods serve different purposes. Holistic evaluation is useful for obtaining a "big picture" image of a translation and quickly determining whether it meets specifications, but cannot provide detailed feedback on specific errors or suggest concrete remedial action. Analytic evaluation is good at identifying and documenting specific problems, but may not capture the overall impression. In addition, it is time-consuming and requires training for evaluators to apply consistently.

Because analytic quality evaluation is resource-intensive, MQM supports both types of evaluation. To use MQM for holistic evaluation, one must create a holistic metric that corresponds to an analytic one. In general, doing so requires the creator

of the metric to select the high-level issues (typically the dimensions) that the analytic metric uses and ask evaluators to consider them for the entire text.

For example, if an analytic metric contains *Grammar*, *Spelling*, and *Typography* under *Fluency*, the corresponding holistic one would ask only about *Fluency.* Rating in such cases typically uses a Likert scale or similar rating mechanism, along the lines of the following:

> On a scale of 0 to 5, where zero indicates that the translation is completely unacceptable and 5 indicates that it is fully acceptable with no detectable problems, how *fluent* is the translation?

Asking these types of questions for each dimension allows the evaluator to form an overall image of how well the translation meets specifications without the need to mark all errors. If the holistic questions do not indicate problems, there is no need to conduct a detailed analytic evaluation except in cases where the consequences of undetected errors would be high. On the other hand, if the translation clearly fails the holistic evaluation, there is also no need to invest the time in a detailed analytic evaluation because the reviewer already knows it would not pass. In this case the reviewer would need to note a few examples of the problems to authenticate them and return the translation for rework.

Only in the case where it is unclear from the holistic evaluation whether the translation is acceptable per the specifications would a full analytic evaluation be advisable, although analytic spot-checking can help ensure that holistic evaluations accurately reflect requirements.

4 The DQF Error Typology

The TAUS DQF is a system developed by TAUS that addresses a variety of approaches to quality assessment, including those aimed specifically at MT, such as measuring post-editor productivity, adequacy/fluency evaluation, readability (see Castilho et al., this volume) and crowdsourced evaluation (see Jiménez-Crespo, this volume). The majority of DQF evaluation methods are out of the scope of this article, which focuses solely on the error typology.

Unlike MQM, which took existing metrics and attempted to harmonise them into one master categorisation, TAUS reached out to LSPs and buyers of translation to ask them for best practices in scorecard-style evaluation. They then used these to develop a simple error typology that was focused on the needs of its localisation-oriented members. Thus, rather than trying to address everything, the DQF Error Typology focused on solving a particularly important area.

The first release of the DQF Error Typology had six error types:

- **Accuracy**. Issues related to the transfer of meaning from source to target-language.
- **Linguistic**. Issues related to the language (rather than the meaning) of the target.

- **Terminology**. Issues related to the use of domain- or organisation-specific approved vocabulary.
- **Style**. Issues related to general or company-specific style.
- **Country Standards**. Issues related to adherence to locale-specific formatting guidelines (e.g. for numbers, addresses, or dates).
- **Layout**. Issues related to non-textual aspects of the content, such as links, formatting, length, and text truncation.

It also featured four additional categories that were used to mark issues that were not errors:

- **Query implementation**. Used to mark changes that needed to be made in response to questions to the content creator.
- **Client edit**. Edits requested by the client.
- **Repeat**. Used to mark cases where an error is repeated, but is done consistently, to avoid penalising the translator for each occurrence.
- **Kudos**. Adds a scoring bonus for something the reviewer feels the translator did well.

Each of the six primary issue types had a number of subtypes: these were initially broken out as their own issue types, but were instead eventually used as examples to explain the six types in question. The DQF Error Typology contained four severity levels, which were assigned numbers. These correspond roughly with the MQM severities.

The first release of the DQF Error Typology was as a scorecard in Excel format. It contained instructions for use and sheets where users could enter error counts for each of the categories with four severity levels to generate a score. This Excel sheet was comparable to what many LSPs used internally for their error-tracking activities.

5 Integrating MQM and DQF

In 2014 reviewers for the QTLaunchPad project pointed out that the development of MQM and DQF along separate tracks threatened to generate market confusion and delay adoption of improved quality practices. They recommended that any applications for future projects that involved translation quality should include integration of the two as a prerequisite for funding. Accordingly, the application for the follow-up project QT21, in which both TAUS and DFKI were partners, included a plan to integrate the two formats. Work began on this project in September 2014 and proceeded through the summer of 2015.

In this work, both formats underwent substantial changes, including the following:

- The dimensions of MQM were restructured to match DQF's top-level categories, meaning that implementers of the DQF error typology would not need to change

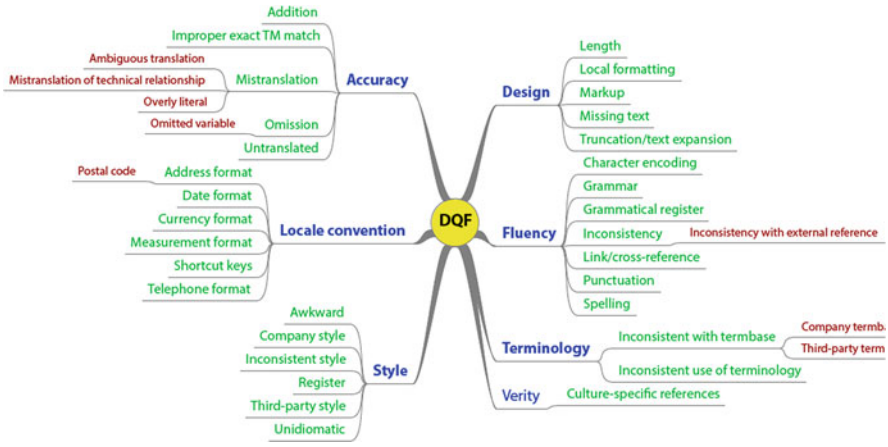

Fig. 4 DQF/MQM integrated error typology

their evaluation criteria. This resulted in the promotion of *Terminology* and *Style* to full dimensions (*Terminology* had previously been split between *Accuracy* and *Fluency*, and *Style* had been included in *Fluency*).

- MQM added a fourth severity level ("null") to match DQF's approach and to allow for issues to be marked without assigning penalties to them.
- DQF adopted the MQM issue names to become a subset of MQM.
- DQF expanded its catalogue of issues to include what had previously been examples so that they were tied to MQM issue types and could be used directly on their own.
- DQF added the *Internationalisation* and *Verity* dimensions to address these issues, which had previously been lumped with other issue types. This shift made the definitions of DQF issues more consistent and clear.

The resulting hierarchy, presented in Fig. 4, contains 50 issue types.

It is thus less than one-third the size of the full MQM hierarchy and is focused more tightly on those issues likely to concern industrial buyers of translation and localisation. As with the full MQM, it is not anticipated that adopters will use all the categories. Instead, they are available as needed. Individuals who had adopted the original DQF Error Typology need only to update the names of their categories (where they differ) to comply with MQM. In keeping with the MQM principle of using only as much detail as is needed and following TAUS' attempts to simplify TQA processes, the recommendation is to use the top-level categories unless there is a need to drill down to greater detail.

Since its release in 2015, the integrated DQF/MQM error typology has become the preferred method to implement MQM. Its smaller size makes it easier to use and grasp. Its inclusion in DQF has helped raise the profile of the MQM approach to TQA.

6 Status and Plans for the Future

Active development work on MQM was conducted at DFKI as part of two European Union-funded projects. With the completion of those projects, this work at DFKI ceased, but TAUS has continued to develop and promote DQF and the error typology, including pushing for its inclusion in translation tools.

The effort to develop MQM further has since been taken up within ASTM Committee F43,[10] which has decided to focus initial efforts exclusively on the DQF subset of MQM. It has the support of both DFKI and TAUS in this effort and has brought together a variety of industry, LSP, and governmental users to ensure widespread applicability. Committee F43 has agreed to keep the error typology free and open to the public, but will develop more detailed guidance that will be sold as a formal standard.

As of 2018, the integrated metric has seen considerable uptake and interest from industry. Trados, the most widely used computer-assisted translation tool, offered MQM starting in 2016. XTM Cloud, an online computer-assisted translation tool and translation management system, has implemented the full MQM typology in its error-checking module. Mozilla has adopted a custom MQM/DQF metric for its localisation needs and other companies such as eBay have publicly announced their adoption of the model. LSPs have moved to MQM, and several technology vendors have announced plans to add support for MQM in the future. DFKI continues to use MQM for research into MT quality.

Overall, the future is bright for MQM and DQF as the new standard approach to assessing translation quality. It provides a way to move past some of the problems of previous methods and to establish TQA on a systematic basis that ties it to the needs of users and to shared best practices that help promote transparency and consistency across applications.

Acknowledgements The author thanks the following individuals: Drs. Hans Uszkoreit and Aljoscha Burchardt (DFKI Berlin), who were integral in the development of MQM; Jaap van der Meer and Attila Görög (TAUS), for their development of DQF and contribution to the integrated MQM/DQF metric; Prof. Alan K. Melby (Brigham Young University Translation Research Group), who contributed greatly to MQM and introduced the notion of specifications. Any errors in this publication are the author's alone and do not reflect on the contributions of these individuals.

[10] ASTM International is a leading organisation in the development and delivery of voluntary consensus standards; its Committee F43 on Language Services and Products was formed in 2010 with the aim of enhancing the quality of language services and products.

References

ASTM (2014) ASTM F2575–14 standard guide for quality assurance in translation. ASTM International, West Conshohocken

Nord C (1997) Translating as a purposeful activity. St. Jerome, Manchester

Sirena D (2004) Mission impossible: improve quality, time and speed at the same time. Globalization Insider 13(2.2). Available via: http://www.translationdirectory.com/article387.htm. Accessed 1 Feb 2017

Snow T (2015) Establishing the viability of the multidimensional quality metrics framework. Dissertation, Brigham Young University. Available via: http://scholarsarchive.byu.edu/etd/5593/. Accessed 1 Feb 2017