IQMT

A Framework for Automatic
Machine Translation Evaluation

Technical Manual v1.3

Jesús Giménez* and Enrique Amigó†

*TALP Research Center, LSI Department
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1–3. 08034, Barcelona

†Departamento de Lenguajes y Sistemas Informáticos
Universidad Nacional de Educación a Distancia
Juan del Rosal, 16. 28040, Madrid

jgimenez@lsi.upc.edu, enrique@lsi.uned.es

26th August 2006

Contents

Abstract

This report[1] presents a tutorial on the IQ$_{MT}$[2] package for Machine Translation Evaluation based on *'Human Likeness'*. IQ$_{MT}$ intends to offer a common workbench on which MT evaluation metrics can be utilized and combined. It provides i) a measure to evaluate the quality of any set of similarity metrics (KING), ii) a measure to evaluate the quality of a translation using a set of similarity metrics (QUEEN), and iii) a measure to evaluate the reliability of a test set (JACK). The IQ$_{MT}$ package is freely available[3] for public use under the GNU Lesser General Public License (LGPL) of the Free Software Foundation. Current version includes a set of 31 metrics from 5 different well-known metric families, and allows users to supply their own metrics. For future releases, we are working on the design of new metrics that are able to capture linguistic aspects of translation beyond lexical ones.

[1]The work reported has been funded by the Spanish Ministry of Science and Technology, projects ALIADO (TIC-2002-04447-C02) and R2D2 (TIC-2003-7180).

[2]IQ$_{MT}$ stands for Inside Qarla Machine Translation Evaluation Framework.

[3]The Perl version 1.3 may freely downloaded at http://www.lsi.upc.edu/ nlp/IQMT.

1 Installation

To configure this module, cd to the directory that contains the README file and type the following:

```
perl Makefile.PL
```

Alternatively, if you plan to install IQ$_{MT}$ somewhere other than your system's perl library directory, you can type something like this:

```
perl Makefile.PL PREFIX=/home/me/perl
```

Then to build you run make.

```
make
```

If you have write access to the installation directories, you may then install by typing:

```
make install
```

Remember to properly set 'path' and PERL5LIB variables:

```
set path = ($path /home/me/IQMT-1.3/bin)
setenv PERL5LIB /home/me/IQMT-1.3/lib:$PERL5LIB
setenv PERL5LIB /home/me/IQMT-1.3/tools/METEOR:$PERL5LIB
```

Notes:

- METEOR requires WordNet 2.0 (available at `http://wordnet.princeton.edu`) You may need to properly set the WNHOME variable. (e.g. setenv WNHOME /usr/local/WordNet-2.0/bin)
- GTM requires java (available at `http://www.java.com`).

2 Introduction

Current approaches to Automatic Machine Translation (MT) Evaluation are mostly based on metrics which determine the quality of a given translation according to its similarity to a given set of reference translations. For long, the commonly accepted criterion defining the quality of an evaluation metric has been its ability to capture 'Human Acceptability', i.e. its level of correlation with human evaluators, usually measured in terms of adequacy and fluency.

By far, the most widely used metric in the recent literature is the perennial **BLEU**, which computes lexical matching accumulated precision for n-grams up to length four [PRWZ01]. However, it presents several defficiencies which cast serious doubts on its usefulness, both for sentence-level error analysis [TSM03] and for system-level comparison [CBOK06].

Other well-known and widely-used metrics are NIST [Dod02], WER [NOLN00], PER [TVN$^+$97], GTM [MGT03], ROUGE [LO04a], and METEOR [BL05], just to name a few. All these metrics take into account information at the lexical level. Therefore, their reliability depends strongly on the number of reference translations available.

Having reached a certain degree of maturity, present MT technology requires nowadays the usage of more sophisticated metrics. There are (at least) three main purposes for which the usage of current automatic MT evaluation metrics is clearly unsatisfactory:

Evaluation of Heterogeneous MT Systems. Comparisons between MT systems based on different paradigms are unfair [CBOK06].

MT-error analysis. Most metrics do not work well at the sentence-level, and even if they do so, they do not provide any information or explanation about the type of errors encountered [TSM03].

MT system development. For a long time, this has been the main reason to trust metrics such as BLEU. However, at this point, metrics have become too shallow. For instance, they systematically fail to capture subtle improvements attained by applying linguistic information [OGK$^+$03]. Current metrics also tend to rank state-of-the-art MT systems unrealistically high, tightly close to human performance (Franz Josef Och (Google), talk at ACL'05 Workshop in MT).

However, little work has been done in order to incorporate information at linguistic levels further than lexical. For instance, metrics such as ROUGE and METEOR may consider stemming. We may also find the WNM metric [BH04], a variant of BLEU which weights n-grams according to their statistical salience estimated out from a

monolingual corpus. Additionally, METEOR may perform a lookup for synonymy in WordNet [Fel98]. But all these are still attempts at the lexical level. To our knowledge, the only attempt so far to exploit information at an upper level has been done by [LG05] who introduced a series of syntax-based features based on syntactic tree matching.

Doubtless the design of a metric that is able to capture all the linguistic aspects that distinguish 'correct' translations from 'incorrect' ones is an ambitious and difficult goal. Instead of building such a sophisticated metric we suggest to follow a 'divide and conquer' strategy, and design a set of specialized metrics, devoted to the evaluation of partial aspects of MT quality. The new challenge is how to combine their outputs into a single measure.

In a recent work, [KS04] tried to combine some aspects of different metrics by applying machine learning techniques to build classifiers that distinguished between human-generated ('good') and machine-generated ('bad') translations. They used features inspired in metrics like BLEU, NIST, WER and PER.

Moreover, we suggest a shift in the 'de facto' accepted criterion. MT quality should be measured in terms of 'Human Likeness' instead of 'Human Acceptability'. First, because 'Human Acceptability' (i.e. correlation with human judges) is not granted, since automatic metrics are based on similarity to human references. Second, because 'Human Likeness' ensures 'Human Acceptability'.

Our approach is based on QARLA [AGPV05], a probabilistic framework originally designed for the evaluation of text summarization systems. QARLA automatically identifies the features that distinguish human translations from automatic ones. It permits metric combinations, without any a-priori weighting of their relative importance. Besides, no training or adjustment of parameters is required, there is no need for human assessments, and it does not depend on the scale properties of the metrics being evaluated. The methodology which is closest to QARLA is ORANGE [LO04b]. However, ORANGE does not permit metric combinations.

As a result of our experience we have developed the IQ_{MT} Framework for MT Evaluation based on 'Human Likeness' [GA06, AGGM06]. The IQ_{MT} Framework is publically available, released under the GNU Lesser General Public License (LGPL) of the Free Software Foundation. It may be freely downloaded at:

`http://www.lsi.upc.edu/~nlp/IQMT`

This tutorial is intended to guide you through the process of configuring and setting up the IQ_{MT} framework as well as to provide a basic methodology for MT Evaluation based on 'Human Likeness' . In

Section 3 the fundamentals of the IQ$_{\text{MT}}$ methodology are presented. The system architecture is described in Section 4. The current set of available metrics is described in Section 5. A case of study on the evaluation of the Europarl Corpus Spanish-to-English translation task is presented in Section 6. Finally, ongoing work is outlined in Section 6.3.

3 Fundamentals

IQ$_{\text{MT}}$ is based on the QARLA Framework [AGPV05]. QARLA uses similarity to models (human references) as a building block. The main assumption is that all human references are equally optimal and, while they are likely to be different, the best similarity metric is the one that identifies and uses the features that are common to all human references, grouping them and separating them from automatic translations.

Therefore, one of the main characterisitics of QARLA that differentiates it from other approaches, is that, besides considering the similarity of automatic translations to human references, QARLA additionally considers the distribution of similarities among human references.

3.1 Measures

The input for QARLA is a set of test cases A (i.e. automatic translations), a set of similarity metrics X, and a set of models R (i.e. human references) for each test case. With such a testbed, QARLA provides three measures:

- **KING**$_{A,R}(X)$, a measure to evaluate the descriptive power of a set of similarity metrics.
- **QUEEN**$_{X,R}(A)$**,** a measure to evaluate the quality of a translation using a set of similarity metrics.
- **JACK**(A, R, X)**,** a measure to evaluate the reliability of a test set.

3.1.1 QUEEN

QUEEN operates under the assumption that a good translation must be similar to all human references according to all metrics. QUEEN is defined as the probability, over $R \times R \times R$, that for every metric in X the automatic translation a is closer to a model than two other models to each other:

$$\text{QUEEN}_{X,R}(a) = Prob(\forall x \in X : x(a, r) \geq x(r', r''))$$

where a is the automatic translation being evaluated, $\langle r, r', r'' \rangle$ are three human references in R, and $x(a, r)$ stands for the similarity of r to a according to the similarity metric x. We can think of the QUEEN measure as using a set of tests (every similarity metric in X) to test the hypothesis that a given translation a is a model. Given $\langle a, r, r', r'' \rangle$,

we test $x(a, r) \geq x(r', r'')$ for each metric x. a is accepted as a model only if it passes the test for every metric. Thus, $\text{QUEEN}_{X,R}(a)$ is the probability of acceptance for a in the sample space $R \times R \times R$. This measure has some interesting properties:

(i) it is able to combine different similarity metrics into a single evaluation measure.

(ii) it is not affected by the scale properties of individual metrics, i.e. it does not require metric normalisation and it is not affected by metric weighting.

(iii) Peers (automatic translations) which are very far from the set of models (human references) all receive QUEEN = 0. In other words, QUEEN does not distinguish between very poor translation strategies.

(iv) The value of QUEEN is maximised for peers that "merge" with the models under all metrics in X.

(v) The universal quantifier on the metric parameter x implies that adding redundant metrics does not bias the result of QUEEN.

However, the main drawback of QUEEN is that it requires the use of multiple references (at least three), when in most cases only a single reference translation is available.

3.1.2 KING

Based on QUEEN, QARLA provides a mechanism to determine the quality of a set of metrics, the KING measure:

$$\text{KING}_{A,R}(X) = Prob(\forall a \in A :$$
$$\text{QUEEN}_{X,R-\{r\}}(r) \geq \text{QUEEN}_{X,R-\{r\}}(a)$$

KING represents the probability that, for a given set of human references R, and a set of metrics X, the QUEEN quality of a human reference is greater than the QUEEN quality of *any* automatic translation in A. Therefore, KING measures the ability of a set of metrics to discern between automatic and human translations.

3.1.3 JACK

Again based on QUEEN, QARLA provides a mechanism to determine the reliability of the test set, the JACK measure:

$$\text{JACK}(A, R, X) = Prob(\exists a, a' \in A :$$
$$\text{QUEEN}_{X,R}(a) > 0 \wedge \ \text{QUEEN}_{X,R}(a') > 0$$
$$\wedge \forall x \in X : x(a, a') \leq x(a, r)$$

i.e. the probability over all human references r of finding a couple of automatic translations a, a' which are (i) close to all human references (QUEEN > 0) and (ii) closer to r than to each other, according to all metrics. JACK measures the heterogeneity of system outputs with respect to human references. A high JACK value means that most references are closely and heterogeneously surrounded by automatic translations. Thus, it ensures that R and A are not biased.

3.2 QARLA for MT

QARLA methodology in 4 steps:

1. compute similarity metrics (using IQ*setup*; See Subsection 4.1)
2. determine the set of metrics with highest descriptive power by maximizing over the KING measure (using IQ*eval*-optimizeKING; See Subsection 4.2).
3. compute MT quality according to the QUEEN measure over the optimal metric set. (using IQ*eval* -doQUEEN; See Subsection 4.2).
4. measure the test set reliability by means of the JACK measure (using IQ*eval* -doJACK; See Subsection 4.2).

3.3 Finding an Optimal Metric Set

The optimal set is defined by the combination of metrics exhibiting the highest KING value. However, exploring all possible combinations might not be viable[4]. IQ*eval* provides an implementation of a simple algorithm which performs an approximate search in order to find a suboptimal set of metrics:

1. Individual metrics are ranked by their KING value.
2. Following that order, metrics are individually added to the set of optimal metrics only if the global KING increases.

[4]There are $2^{31} - 1$ possible combinations if we take into account all lexical metrics; See Subsection 5.1.

Although fairly simple, this algorithm provides excellent results in practice. However, we are experimenting new methods for metric set optimization based on Clustering techniques.

4 System Architecture

A schematic plot of the system architecture may be seen in Figure 1. IQ$_{\text{MT}}$ consists of two main components, namely IQ*setup* and IQ*eval*. The IQ*setup* component is responsible for applying a set of similarity metrics to a set of automatic translations and a set of human references. The IQ*eval* component computes the KING, QUEEN, and JACK measures on top of the similarity scores generated by IQ*setup*.

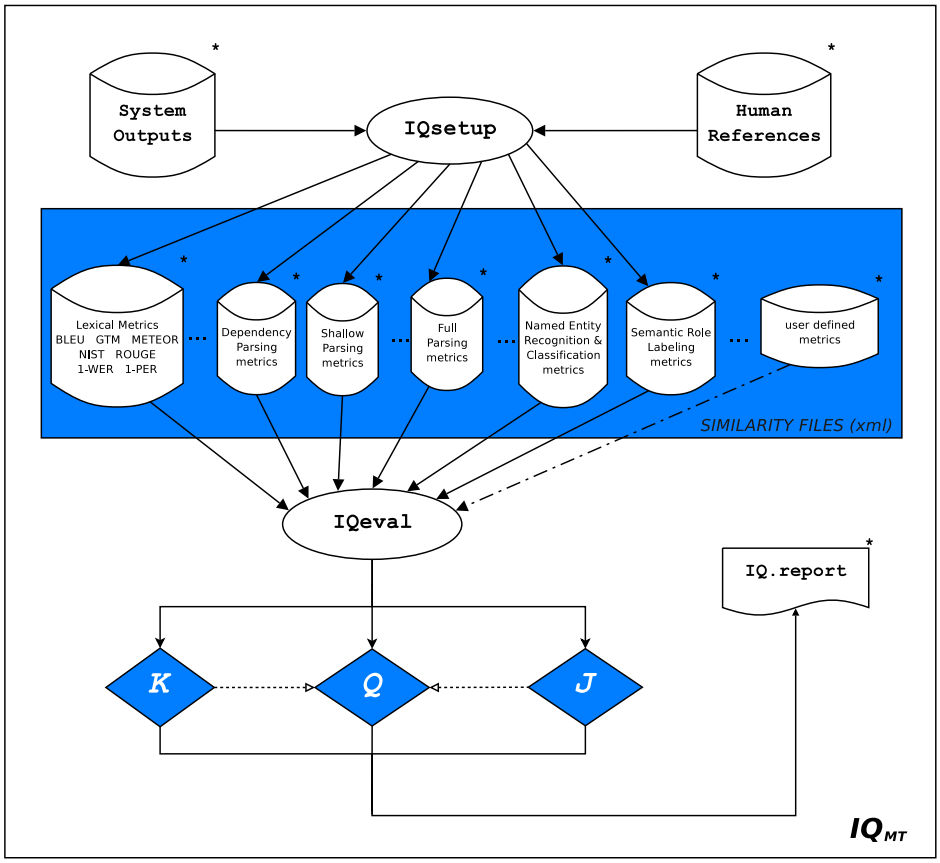

Figure 1: IQ$_{\text{MT}}$ system architecture.

4.1 IQ*setup*

IQ*setup* computes the similarities required for the estimation of the QUEEN measure. This component receives as input a configuration file specifying:

- set of human references (R)

- set of system outputs (i.e. automatic translations) (A)
- set of metrics (X)
- source file (source translation)
- IQ$_{\text{MT}}$ package location (path)

Based on this information, IQ*setup* generates for each metric a collection of *'IQ XML'* similarity files:

- <system>/<system>/<metric>.xml
- <system>/<reference+>/<metric>.xml
- <reference>/<reference+>/<metric>.xml

Source, reference and system files all must contain raw text and follow a 'one sentence per line' format. Therefore, the number of lines in these files must match.

The user must indicate which of the available metrics must be computed:

- doBLEU [BLEU-1 | BLEU-2 | BLEUi-2 | BLEU-3 | BLEUi-3 | BLEU-4 | BLEUi-4]
- doNIST [NIST-1 | NIST-2 | NISTi-2 | NIST-3 | NISTi-3 | NIST-4 | NISTi-4 | NIST-5 | NISTi-5]
- doGTM [GTM-1 | GTM-2 | GTM-3]
- doMETEOR [MTR-exact | MTR-stem | MTR-wnstm | MTR-wnsyn]
- doROUGE [RG-1 | RG-2 | RG-3 | RG-4 | RG-L | RG-W-1.2 | RG-S* | RG-SU*]

For instance, if the user specifies 'doBLEU BLEU-3 BLEU-4' and 'doGTM GTM-2' only three metric variants will be computed, namely BLEU-3, BLEU-4 and GTM-2. If the user specifies 'doBLEU' and 'doGTM' ten variants will be computed, namely BLEU-1, BLEU-2, BLEU-3, BLEU-4, BLEUi-2, BLEUi-3, BLEUi-4, GTM-1, GTM-2 and GTM-3. See an example[5] of IQ*setup* config file in Table 1.

You may then run IQ*setup*:

```
IQsetup IQsetup.config IQeval.config
```

Options are:

[5]Lines begining with '#' are comments.

```
# – EXPERIMENT NAME
NAME=MT_DUMMY_TESTSET
# – IQMT LOCATION
IQMT=/home/users/me/IQMT/
# – FILES
source=source_file.txt
ref=reference_file.txt.1
...
ref=reference_file.txt.M
system=system_output_file.txt.1
...
system=system_output_file.txt.N
# – AVAILABLE METRICS
doBLEU
doNIST
doGTM
doMETEOR
doROUGE
# doBLEU BLEUi-2 BLEU-2 BLEU-4
# doNIST NISTi-2 NISTi-3 NIST-2 NIST-5
# doGTM GTM-1 GTM-2
# doMETEOR MTR-exact MTR-stem MTR-wnstm MTR-wnsyn
# doROUGE RG-1 RG-2 RG-3 RG-4 RG-L RG-W-1.2 RG-S* RG-SU*
```

Table 1: IQ*setup* configuration file.

-JACK This option enables computation of <system>/<system>/<metric>.xml files, which are not computed by default.

-remake This option forces recomputation of existing similarity files, which are not recomputed by default.

4.1.1 *'IQ XML'* Representation Schema

The *'IQ XML'* schema is intended unify the representation of evaluation scores at the sentence level.

```
<IQ metric="BLEU-4" ref="R0" score="0.3945" target="S0">
  <S n="1">0.3033</S>
  <S n="2">0.5833</S>
  ...
  <S n="1007">0.6852</S>
```

```
  <S n="1008">0.8333</S>
</IQ>
```

For instance, the file above provides system and segment (i.e. sentence) level similarity scores obtained by comparing system 'S0' against reference 'R0' based on the 'BLEU-4' similarity metric.

But the main advantage of the *'IQ XML'* representation schema is that it allows users to supply their own metrics in a transparent and unified manner (See Subsubsection 4.1.2). For every new metric, the user is responsible for generating an *IQ XML* similarity file for each pair <system-reference+>, <reference-reference+>, and <system-system>.

4.1.2 Playing with your own metrics

IQ$_{MT}$ allows the user to supply their own metrics through the 'IQ *XML'* schema of data representation (See Subsubsection 4.1.1).

Filenames are important. They must follow this format:

- **TARGET**/**REFERENCE**/**metric**.xml.

The user must provide an XML file for each pair of:

- REFERENCE-REFERENCE+
- SYSTEM-REFERENCE+
- SYSTEM-SYSTEM (only in the case of the JACK measure)

Similarities when TARGET and REFERENCE are the same item are not necessary. For instance, suppose you have a working set consisting of two systems ('S0' and 'S1') and three references ('R0', 'R1' and 'R2'). If you add a new metric called 'NEWMETRIC', you must supply 15 XML files:

- R0/R1/NEWMETRIC.xml
- R0/R2/NEWMETRIC.xml
- R1/R0/NEWMETRIC.xml
- R1/R2/NEWMETRIC.xml
- R2/R0/NEWMETRIC.xml
- R2/R1/NEWMETRIC.xml
- S0/R0/NEWMETRIC.xml
- S0/R1/NEWMETRIC.xml
- S0/R2/NEWMETRIC.xml

- S1/R0/NEWMETRIC.xml
- S1/R1/NEWMETRIC.xml
- S1/R2/NEWMETRIC.xml

That works for the QUEEN and KING components. If the JACK measure for test set reliability is desired 4 additional XML files must be supplied:

- S0/S1/NEWMETRIC.xml
- S1/S0/NEWMETRIC.xml
- S2/S0/NEWMETRIC.xml
- S2/S1/NEWMETRIC.xml

Moreover, if you plan to use the "-doOQ" option with the new metric, remember to provide results outside QARLA for all the systems in a multiple reference setting:

- SYSTEM-REFERENCE'0_..._REFERENCE'N

Again, filenames are important:

- **TARGET**/**REF**$_0$_..._**REF**$_i$..._**REF**$_N$/**metric**.xml

In our example, you should provide two extra files:

- S0/R0_R1_R2/NEWMETRIC.xml
- S1/R0_R1_R2/NEWMETRIC.xml

Finally, remember to properly edit the IQ*eval* config file, so you can play with your new metric:

```
metrics_NEWMETRIC= NEWMETRIC

metrics=BLEU-1 BLEU-2 BLEU-3 BLEU-4 BLEUi-2 BLEUi-3 BLEUi-4
        GTM-1 GTM-2 GTM-3 MTR-exact MTR-stem MTR-wnstm
        MTR-wnsyn NIST-1 NIST-2 NIST-3 NIST-4 NIST-5 NISTi-2
        NISTi-3 NISTi-4 NISTi-5 RG-1 RG-2 RG-3 RG-4 RG-L
        RG-SUs RG-Ss RG-W-1.2 NEWMETRIC
```

4.2 IQ*eval*

IQ*eval* allows us to calculate the KING, QUEEN and JACK measures.

-doKING : compute KING score(s).

-doQUEEN : compute QUEEN score(s).

-doJACK : compute JACK score.

Other **actions** are available:

-doOQ : compute individual MT evaluation scores outside QARLA.

-optimizeKING : perform metric set optimization based on KING (See Subsection 3.3).

-doPking | -doPorange : compute P_{KING} or P_{ORANGE} probabilitites, as described in [AGGM06].

-doLQUEEN | -doLKING : compute QUEEN or KING score(s) for each metric individually.

-TRY | -TRYoq | -TRYall : add a new system and compute QUEEN or metrics outside QARLA or both.

Several **options** may be specified:

-R `<set_name>` : the set of references. All references are used by default.

-S `<set_name>` : the set of system outputs to evaluate. All systems are evaluated by default.

-M `<set_name>` : the set of metrics. All metrics are considered by default.

-T `<set_name>` : the subset of sentences per system to evaluate. All sentences are considered by default.

-G `<granularity>` : return scores at the sentence ('-G seg') / system ('-G sys') level.

-TT : enable trans-topic mode for KING/QUEEN/JACK computations.

-doref : include reference scores.

-remake : remake metric computations.

-O `<output_format>` : output may be presented as:

score matrix : ('-O 0') where each column corresponds to a metric, and each row corresponds to a system / segment depending on the level of granularity.

ranking lists : ('-O 1') each column (results corresponding to the same metric) is listed separatedly.

[sigrona] /home/users/me/IQMT > IQeval -doOQ -G sys -O 0 IQeval.config

SYS	BLEU-4	GTM-2	MTR-wnsyn	NIST-5	RG-L	QUEEN
S0	0.6232	0.4058	0.7744	11.3452	0.6675	0.4369
S1	0.6453	0.4177	0.7882	11.6098	0.6776	0.4819
S2	0.5684	0.3829	0.7387	10.6599	0.6411	0.3465
S3	0.6256	0.4091	0.7728	11.4734	0.6715	0.4509
S4	0.5901	0.3922	0.7415	10.8246	0.6473	0.3618
S5	0.6472	0.4171	0.7725	11.6038	0.6767	0.4737

Table 2: Running IQ*eval*.

Set names are specified according to the names provided in a configuration file, which is automatically generated by the IQ*setup* component, as a by-pass product. This configuration file contains a series of predefined sets. It must be edited in order to define new sets.

See an example of IQ*eval* output in Table 2.

A specific set of metrics / systems / references / segments may be used:

- BLEU-4 and NIST-5 metrics
- systems S0 and S1
- references R0, R1 and R2
- segments [1, 2, 3, 10, 50..100, 200..250, 300, 310, 400-500]

You would have to define these sets in the IQeval.config file, for instance:

```
some_metrics= BLEU-4 NIST-5
some_systems= S0 S1
some_refs= R0 R1 R2
some_segs= 1-3, 10, 50-100, 200-250, 300, 310, 400-500
```

and then, rerun IQ*eval* (see Table 3). The granularity level has been changed ('-G seg') to see the effect of the segment selection.

```
[sigrona] /home/users/me/IQMT > IQeval -doOQ
                                -doQUEEN -G seg -O 0 -M some_metrics
                                -S some_systems -R some_refs
                                -T some_segs IQeval.config
```

SYS	BLEU-4	NIST-5	QUEEN
S0:1	0.0000	7.6320	0.4444
S0:2	0.6851	12.8007	0.6111
S0:3	0.0000	6.9161	0.0000
S0:10	0.5990	10.8767	0.8889
S0:50	0.5731	12.7768	0.5000
S0:51	0.4431	9.8990	0.1111
...			
S0:499	0.7698	11.2825	0.4444
S0:500	0.5221	10.5259	0.2778
S1:1	0.0000	7.6320	0.4444
S1:2	0.6851	12.8007	0.6111
S1:3	0.0000	9.0135	0.0000
S1:10	0.5612	10.9241	0.8889
S1:50	0.5731	12.7768	0.5000
S1:51	0.8743	14.3287	0.5556
...			
S1:499	0.7044	10.9209	0.4444
S1:500	0.5514	10.7646	0.4444

Table 3: Running IQ*eval*.

5 Similarity Metrics

The set of similarity metrics is a dynamic component in our framework. We have started by adapting existing MT evaluation metrics. These metrics are transformed into similarity metrics by considering just a single reference when computing its value.

However, our main target is to develop a set of metrics that capture linguistic information at levels of abstraction futher than lexical, i.e. syntactic and (shallow-)semantic.

5.1 Lexical Metrics

IQ_{MT} currently allows the usage of a number of existing automatic MT evaluation metrics such as BLEU, NIST, GTM, ROUGE, and METEOR. 31 variants of these 5 families of metrics have been integrated and tested so far[6]:

[6]WER and PER metrics have been also tested, but could not be released for copyright reasons.

BLEU-n | BLEUi-n: [7] accumulated and individual BLEU scores for several n-gram levels ($n = 1...4$).

NIST-n | NISTi-n: [8] accumulated and individual NIST scores for several n-gram levels ($n = 1...5$).

GTM-e: [9] for several values of the e parameter ($e = 1...3$).

METEOR: [10] We use 4 variants:

- **MTR-exact:** running 'exact' module.
- **MTR-porter:** (default) running 'exact' and 'porter_stem' modules, in that order.
- **MTR-wnstm:** running 'exact', 'porter_stem' and 'wn_stem' modules, in that order.
- **MTR-wnsyn:** running 'exact', 'porter_stem', 'wn_stem' and 'wn_synonymy' modules, in that order.

ROUGE: [11] [LO04a]:

- **RG-n** for several n-grams ($n = 1...4$)
- **RG-L:** longest common subsequence (LCS).
- **RG-S*:** skip bigrams with no max-gap-length.
- **RG-SU*:** skip bigrams with no max-gap-length, including unigrams.
- **RG-W:** weighted longest common subsequence (WLCS) with weighting factor $w = 1.2$.

mWER: we use $1 - \text{mWER}$.

mPER: we use $1 - \text{mPER}$.

[7] We use mteval-kit-v10/mteval-v11b.pl for BLEU.

[8] We use mteval-kit-v10/mteval-v11b.pl for NIST.

[9] We use GTM version 1.2.

[10] We use METEOR version 0.4.3.

[11] We used ROUGE version 1.5.5. Options are '-z SPL -2 -1 -U -m -r 1000 -n 4 -w 1.2 -c 95 -d'.

6 A case of study: Europarl

For a robust estimation of the KING, QUEEN, and JACK probabilities, the ideal scenario would consist of a large number of human references per sentence, and automatic outputs generated by heterogeneous MT systems. Unfortunately, this kind scenario is rarely found. Generally, few references are available (one in most cases), and MT systems are similar to each other. Thus, we have tested our system under a more realistic scenario. We utilize the data from the *'Openlab 2006'* Initiative[12] promoted by the TC-STAR[13] Consortium.

6.1 Experimental Setting

'Openlab 2006' data are entirely based on European Parliament Proceedings[14], covering April 1996 to May 2005. We focus on the Spanish-to-English translation task. The training set consists of 1,281,427 parallel sentences. For evaluation purposes we use the development set which consists of 1,008 sentences. Three human references per sentence are available. We intend to evaluate 4 systems:

- Word-based SMT system (WB).
- Systran Rule-based translation engine (SYSTRAN).
- Phrase-based SMT system (PB).
- Phrase-based SMT system (PB++)[15].

SMT systems are built as described in [GM05]. As to 'SYSTRAN', we used the freely available on-line version[16]. Let us note that evaluation is unfair to 'SYSTRAN' because SMT systems have been trained using in-domain data. However, we include 'SYSTRAN' for the sake of heterogeneity. We use a set of 26 metric variants[17].

6.2 Evaluating with Standard Metrics

First we analyze the individual behaviour of standard metrics outside QARLA. See results in Table 4. We use one representative from each family, the metric variant with highest KING value in the given test

[12] http://tc-star.itc.it/openlab2006/

[13] http://www.tc-star.org/

[14] http://www.europarl.eu.int/

[15] This system is an improved version of the 'PB' system which uses information at the shallow-parsing level to build better translation models [GM05].

[16] http://www.systransoft.com.

[17] Only individual BLEU and NIST scores are not used.

System	1-PER	1-WER	BLEU-3	GTM-2	MTR	NIST-3	RG-L
WB	0.66	0.58	0.50	0.33	0.57	8.79	0.56
SYSTRAN	0.70	0.60	0.56	0.36	0.65	9.59	0.63
PB	**0.74**	**0.64**	**0.66**	**0.41**	0.69	10.66	0.66
PB++	**0.74**	0.63	**0.66**	**0.41**	**0.70**	**10.72**	**0.67**

Table 4: MT quality according to several metrics outside QARLA.

set. Results indicate that Phrase-based systems (*'PB'* and *'PB++'*) are best according to all metrics, attaining very similar scores. However, there is not agreement between metrics in order to decide which system between these two is best. Three metrics reflect a tie ('1-PER', 'BLEU' and 'GTM-2'), three other metrics score the *'PB++'* system higher ('MTR-EXACT', 'NIST-3' and 'RG-L'), and only one metric ranks the *'PB'* system first ('1-WER'). Although differences are minor, the key question is "which metric should I trust?".

Interestingly, note that, contrary to our expectations, the 'SYSTRAN' system outperforms the word-based system according to all metrics.

6.3 Evaluating with IQ_{MT}

Inside the IQ_{MT} Framework systems are evaluated according to their 'Human Likeness'. Thus, we must trust the metric (or set of metrics) with highest descriptive power (highest KING), i.e. the metric which best identifies the features that distinguish between human translations and automatic translations. Table 5 shows the KING value for each individual metric.

In this test set, metrics from the NIST family consistently obtain the highest KING values, ranging from 0.34 to 0.37. Only the '1-WER' metric achieves a comparable descriptive power (KING = 0.34).

We apply the algorithm described in Subsection 3.3. In the case of the *'Openlab 2006'* data, we can count only on three human references per sentence. In order to increase the number of samples for QUEEN estimation we can use reference similarities $x(r', r'')$ between manual translation pairs from other sentences, assuming that the distances between manual references are relatively stable across examples. The optimal set is:

{NIST-2, NIST-3, NIST-4, and 1-WER}

It attains a KING measure of 0.38, which means that in 38% of the cases this metric set is able to identify human references with respect

Evaluation metric	**KING**
1-PER	**0.30**
1-WER	**0.34**
BLEU-1	0.29
BLEU-2	0.32
BLEU-3	**0.32**
BLEU-4	0.32
GTM-1	0.30
GTM-2	**0.32**
GTM-3	0.31
MTR-exact	**0.29**
MTR-stem	0.28
MTR-wnstm	0.28
MTR-wnsyn	0.29
NIST-1	0.34
NIST-2	0.37
NIST-3	**0.37**
NIST-4	0.37
NIST-5	0.36
RG-1	0.29
RG-2	0.32
RG-3	0.32
RG-4	0.31
RG-L	**0.33**
RG-SUs	0.32
RG-Ss	0.32
RG-W-1.2	0.29

Table 5: Descriptive power of standard metrics (KING).

to all automatic translations. Interestingly, the optimal set contains metrics working at all levels of granularity from 1-grams to 4-grams.

MT System	**QUEEN**
WB	0.31
SYSTRAN	0.39
PB	0.45
PB++	0.46

Table 6: MT quality according to the optimal metric set inside the IQ_{MT} Framework (QUEEN measure).

We use this metric set to compute the QUEEN measure for all systems. See results at the system level in Table 6. As expected, phrase-based systems attain best results, significantly better than the word-based system and 'SYSTRAN'. 'PB++' slightly outperforms 'PB', although not very significantly. Interestingly, the 'SYSTRAN' system performs significantly better than the 'WB' system. This means that, in this test set, translations produced by 'SYSTRAN' are more human-like than those produced by the word-based SMT system, even though 'SYSTRAN' is not designed for the specific domain.

Moreover, the QUEEN measure at the sentence level allows the user to perform a detailed error analysis by inspecting particular cases. Table 7 shows an interesting case of error analysis, in which all systems attain a QUEEN score under 0.2 except the 'PB++' system which scores 0.83. The QUEEN measure identifies the features which characterize human translations. QUEEN favours those automatic translations which share these features that are common to all references. In this case the 'PB++' system output is rewarded for providing exact translations, according to all references, for *'gestión de las crisis'* ('crisis management') and *'esperan señales'* ('they expect signs'). On the other hand, the automatic translations which do not share these common features are penalized.

Finally, the quality of the given test set of systems, references and metrics (JACK measure), considering the optimal metric set, is 0.77. This means that, in most cases (77%), system outputs are heterogeneously distributed closely around human references according to all metrics, and consequently, the test set is representative and reliable.

source	los ciudadanos esperan de nosotros algo más que la simple **gestión de las crisis ; esperan señales** y una política sostenible en estos ámbitos .

systems

WB	the citizens expect of us something more than the simple **management of the crisis** and a sustainable policy in these areas . **expectantly signals**
SYSTRAN	the citizens wait for of us something more than the simple **management of the crises; they wait for signals** and a sustainable policy in these scopes.
PB	the citizens expect us any more than simply **managing crises ; they hope signals** and a sustainable policy in these areas .
PB++	the citizens expect us something more than simply **crisis management ; they expect signs** and a sustainable policy in these areas .

references

R0	the public expect more than just **crisis management ; they expect signs** , and a sustainable policy in these fields .
R1	citizens expect something more of us than just simple **crisis management ; they expect signs** and sustainable policies in these areas .
R2	the citizens expect from us something more than a simple **crisis management ; they expect signs** and a sustainable policy in these matters .

Table 7: A case of error analysis, according to the QUEEN measure, in which the 'PB++' system outperforms the rest.

Ongoing work

Currently, we are devoting our main efforts to the development, study, and integration inside the QARLA Framework, of new families of partial metrics at the lexical, sintactic and shallow semantic levels.

Feedback

Discussion on this software as well as information about oncoming updates takes place on the IQ$_{\text{MT}}$ google group, to which you can subscribe at:

http://groups-beta.google.com/group/IQMT

and post messages at IQMT@googlegroups.com.

References

[AGGM06] Enrique Amigó, Jesús Giménez, Julio Gonzalo, and Lluís Màrquez. Mt evaluation: Human-like vs. human acceptable. In *Proceedings of COLING-ACL06*, 2006.

[AGPV05] Enrique Amigó, Julio Gonzalo, Anselmo Peñas, and Felisa Verdejo. Qarla: a framework for the evaluation of automatic sumarization. In *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics*, 2005.

[BH04] Bogdan Babych and Tony Hartley. Extending the bleu mt evaluation method with frequency weightings. In *Proceedings of ACL*, 2004.

[BL05] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005.

[CBOK06] Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of EACL*, 2006.

[Dod02] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the 2nd Internation Conference on Human Language Technology*, pages 138–145, 2002.

[Fel98] C. Fellbaum, editor. *WordNet. An Electronic Lexical Database*. The MIT Press, 1998.

[GA06] Jesús Giménez and Enrique Amigó. IQMT: A Framework for Automatic Machine Translation Evaluation. In *Proceedings of the 5th LREC*, 2006.

[GM05] Jesús Giménez and Lluís Màrquez. Combining linguistic data views for phrase-based smt. In *Proceedings of the Workshop on Building and Using Parallel Texts, ACL*, 2005.

[KS04] Alex Kulesza and Stuart M. Shieber. A learning approach to improving sentence-level mt evaluation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, 2004.

[LG05] Ding Liu and Daniel Gildea. Syntactic features for evaluation of machine translation. In *Proceedings of ACL Work-*

shop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005.

[LO04a] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statics. In *Proceedings of ACL*, 2004.

[LO04b] Chin-Yew Lin and Franz Josef Och. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of COLING*, 2004.

[MGT03] I. Dan Melamed, Ryan Green, and Joseph P. Turian. Precision and recall of machine translation. In *Proceedings of HLT/NAACL*, 2003.

[NOLN00] S. Nießen, F.J. Och, G. Leusch, and H. Ney. Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, 2000.

[OGK+03] Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. Final report of johns hopkins 2003 summer workshop on syntax for statistical machine translation. Technical report, Johns Hopkins University, 2003.

[PRWZ01] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation, ibm research report, rc22176. Technical report, IBM T.J. Watson Research Center, 2001.

[TSM03] Joseph P. Turian, Luke Shen, and I. Dan Melamed. Evaluation of machine translation and its evaluation. In *Proceedings of MT SUMMIT IX*, 2003.

[TVN+97] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf. Accelerated dp based search for statistical translation. In *Proceedings of European Conference on Speech Communication and Technology*, 1997.