IQ$_{MT}$: A Framework for Machine Translation Evaluation v 1.0 Technical Manual.

Jesús Giménez Enrique Amigó

30th December 2005

1. Introduction

Current approaches to Machine Translation (MT) Evaluation are clearly unsatisfactory. Most of the existing metrics work only at the lexical level, by rewarding n-gram matches between an automatic translation and a set of human references.

Without a single doubt, the construction of a metric that is able to capture all the linguistic aspects that distinguish 'correct' translations from 'incorrect' ones is a very difficult path to trace.

In our work we approach this challenge by following a 'divide and conquer' strategy. We suggest to build a set of specialized metrics each one devoted to the evaluation of a concrete aspect. The point then is how to combine a set of metrics into a single measure of MT quality.

The IQ_{MT} framework is based on the QARLA framework (Amigó et al., 2005). It permits metric combinations, with the singularity that there is no need to perform any training or adjustment of parameters. Inside IQ_{MT} individual metrics improve their performance with respect to the system-level correlation both in adequacy and fluency with human assessments. However, our main target is to develop a set of metrics that capture linguistic information at different levels of abstraction: lexical, syntactic and semantic.

This tutorial is intended to guide you through the process of configuring and setting up the IQ_{MT} framework. In Section 2. the fundamentals of the IQ_{MT} methodology are presented. The system architecture is described in Section 3.. Finally, Section 4. explains in detail how to use your own metrics inside IQ_{MT}.

2. Fundamentals

IQ_{MT} uses similarity to human references as a building block. Several metrics may be combined in a single measure, IQ, based on the QUEEN measure suggested in QARLA (Amigó et al., 2005). The IQ measure operates under the assumption that a good translation must be at least as similar to one of the references as the rest of references are to each other, according to all metrics in a given set.

We define the IQ measure. Given a set of similarity metrics X, and a set of references R for each test case, if a translation t is equal to one reference, then $IQ_X(t, R)$ is maximum. For this, we consider the distance from t to the nearest reference in R:

$$IQ_X(a, R) \equiv max_{r \in R}\, iq_{X,R}(a, r)$$

$$iq_{X,R}(a,r) = \begin{cases} 1 & \text{if } \forall x \in X : \forall r', r'' \in R : \\ & \qquad x(a,r) \geq x(r',r'') \\ 0 & otherwise \end{cases}$$

Therefore, at the sentence level IQ behaves as a binary measure which tells whether a given translation t is correct (it satisifies the criterion above) or not.

This measure exhibits the same properties than its predecessor QUEEN:

(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 all receive IQ=0. In other words, IQ does not distinguish between very poor translation strategies.

(iv) The value of IQ is maximised for peers that “merge” with the models (human references) under all metrics in X.

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

Further details may be found in (Giménez et al., 2005).

3. System Architecture

The system architecture may be seen in Figure 1. IQ$_{\text{MT}}$ has two main components, namely IQ*setup* and IQ*eval*. The IQ*setup* component is responsible for applying a set of metrics to a set of translations and a set of references. The IQ*eval* component computes IQ scores on top of the scores generated by IQ*setup*.

3.1. IQ*setup*

The IQ*setup* component is responsible for applying a given set of metrics to a given set of translations by different systems.

IQ$_{\text{MT}}$ currently allows the usage of a number of existing automatic MT evaluation metrics such as BLEU, NIST, GTM, ROUGE, and METEOR. 24 variants of these 5 families of metrics have been integrated and tested so far[1]:

[1] WER and PER (Tillmann et al., 1997) metrics have been also tested, but could not be released for reasons of copyright.

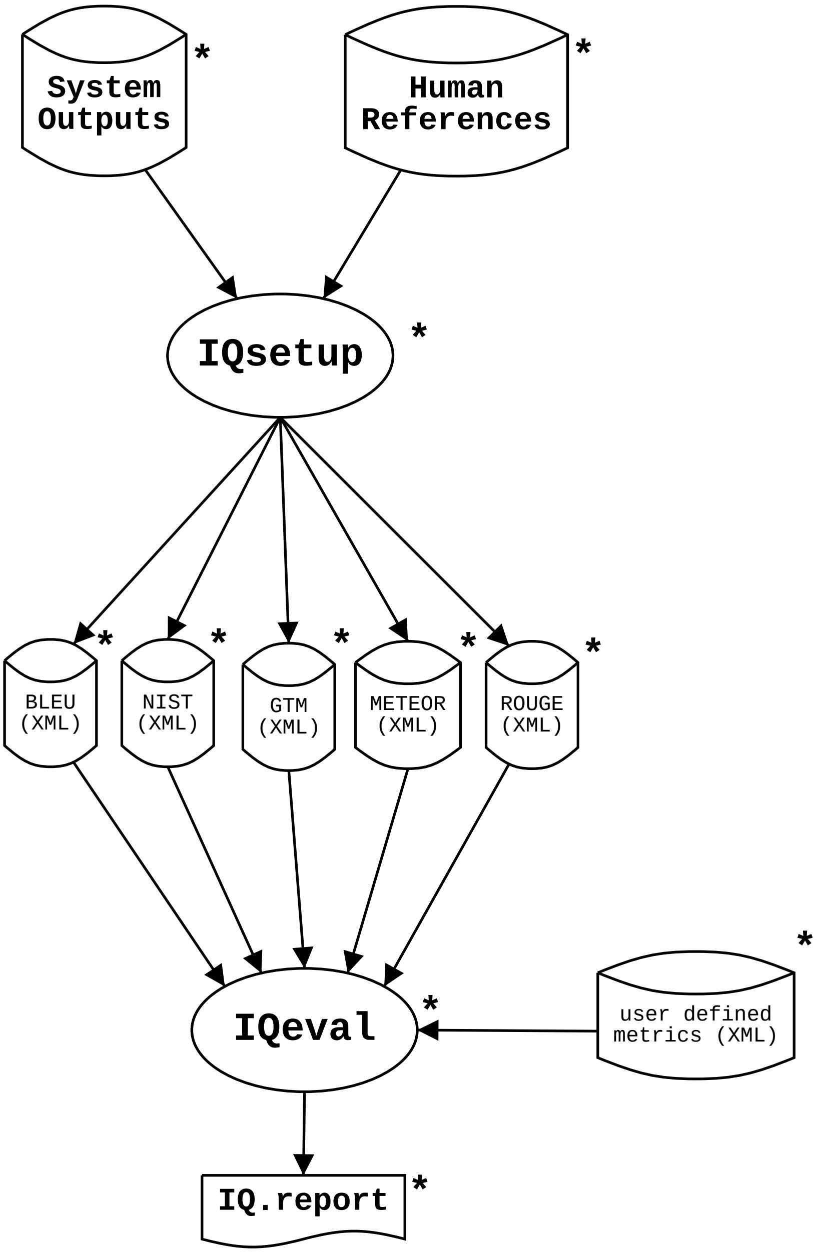

Figure 1: IQ<span>MT</span> system architecture.

BLEU [2] (Papineni et al., 2001) accumulated BLEU scores for several n-gram levels ($n = 1, 2, 3, 4$).

[2] We used mteval-kit-v10/mteval-v11b.pl for BLEU calculation.

NIST [3] (Doddington, 2002) accumulated NIST scores for several n-gram levels ($n = 1, 2, 3, 4, 5$).

GTM [4] for several values of the e parameter ($e = 1, 2, 3$) (Melamed et al., 2003).

METEOR [5] (Banerjee and Lavie, 2005) We used 4 variants.

METEOR.exact running "exact" module only.

METEOR.porter (default) running "exact" and "porter_stem" modules, in that order.

METEOR.wn1 running "exact", "porter_stem" and "wn_stem" modules, in that order.

METEOR.wn2 running "exact", "porter_stem", "wn_stem" and "wn_synonymy" modules, in that order.

ROUGE [6] (Lin and Och, 2004) for several n-grams ($n = 1, 2, 3, 4$), and 4 other variants at the 4-gram level:

ROUGE-L longest common subsequence (LCS).

ROUGE-S* skip bigrams with no max-gap-length.

ROUGE-SU* skip bigrams with no max-gap-length, including uningrams.

ROUGE-W weighted longest common subsequence (WLCS) with weighting factor $w = 1.2$.

The IQ*setup* component requires a config file which must specify several variables:

- source file (source translation)
- system files (set of target translations)
- reference files (set of human reference translations)
- set of metrics
- IQMT location (path)

[3] We used mteval-kit-v10/mteval-v11b.pl for NIST calculation.

[4] We used GTM version 1.2.

[5] We used METEOR version 0.4.3.

[6] 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’’`.

Source, reference and system files all must contain raw text and follow a 'one sentence per line' format. The user must indicate which of the available metrics must be computed, if any:

- doBLEU [1 |2 |3 |4]
- doNIST [1 |2 |3 |4 |5]
- doGTM [1 |2 |3]
- doMETEOR [exact |stem |wnstm |wnsyn]
- doROUGE [1 |2 |3 |4 |L |W |S |SU]

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

You may then run IQ*setup*

```
Usage : IQsetup  [options]  <IQsetup.config>  <IQeval.config>
                                (input)          (output)

  - print         : print similarities onto IQeval.config
                    (default disabled)
  - remake        : remake metric computations
  - V <0|1|2>     : verbosity
                    0 - non-verbose (default)
                    1 - low verbosity
                    2 - medium verbosity

Example: IQsetup IQsetup.config IQeval.config
```

Given the 'setup' config file, IQ*setup* generates an 'evaluation' config file in a format convenient for the IQ*eval* component, and a series of XML files contaning MT evaluation scores for each metric and each pair:

- SYSTEM*-REFERENCE*
- REFERENCE*-SYSTEM*
- SYSTEM*-SYSTEM*
- REFERENCE*-REFERENCE*

```
# – EXPERIMENT NAME
NAME=IWSLT04_CE
# – 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 1 2 3 4
# doNIST 1 2 3 4 5
# doGTM 1 2 3
# doMETEOR exact stem wnstm wnsyn
# doROUGE 1 2 3 4 L W S SU
```

Table 1: IQ*setup* configuration file.

3.2. IQ*eval*

Given a table of similarities, it allows to calculate IQ scores. Several options are currently available:

metrics set of metrics to use.

systems set of systems to evaluate.

references set of references to use.

segments set of translations to use.

granularity return scores at the sentence (‘-G seg’) / system (‘-G sys’) level.

output format output may be presented as:

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

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

IQ*eval* allows also to obtain individual scores for each of the metrics in the given set as they are outside the IQ$_{\text{MT}}$ framework. See an example of IQ*eval* output in Table 2.

```
[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       IQ

S0    0.6232   0.4058     0.7744   11.3452   0.6675   0.4369   0.3452
S1    0.6453   0.4177     0.7882   11.6098   0.6776   0.4819   0.4107
S2    0.5684   0.3829     0.7387   10.6599   0.6411   0.3465   0.2520
S3    0.6256   0.4091     0.7728   11.4734   0.6715   0.4509   0.3810
S4    0.5901   0.3922     0.7415   10.8246   0.6473   0.3618   0.2579
S5    0.6472   0.4171     0.7725   11.6038   0.6767   0.4737   0.3988
```

Table 2: Running IQ*eval*.

Now suppose you want to use a specific set of metrics / systems / references / segments. For instance, you want to use only:

- 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]

The 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') too see the effect of the segment selection.

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

```
SYS       BLEU-4   NIST-5   QUEEN       IQ

S0:1      0.0000   7.6320   0.4444   0.0000
S0:2      0.6851  12.8007   0.6111   1.0000
S0:3      0.0000   6.9161   0.0000   0.0000
S0:10     0.5990  10.8767   0.8889   1.0000
S0:50     0.5731  12.7768   0.5000   1.0000
S0:51     0.4431   9.8990   0.1111   0.0000
...
S0:499    0.7698  11.2825   0.4444   0.0000
S0:500    0.5221  10.5259   0.2778   0.0000
S1:1      0.0000   7.6320   0.4444   0.0000
S1:2      0.6851  12.8007   0.6111   1.0000
S1:3      0.0000   9.0135   0.0000   0.0000
S1:10     0.5612  10.9241   0.8889   1.0000
S1:50     0.5731  12.7768   0.5000   1.0000
S1:51     0.8743  14.3287   0.5556   1.0000
...
S1:499    0.7044  10.9209   0.4444   0.0000
S1:500    0.5514  10.7646   0.4444   0.0000
```

Table 3: Running IQ*eval*.

4. Playing with your own metrics

The main feature of IQMT is that it allows to robustly combine different metrics, possibly working at different linguistic levels. In order to allow the user to introduce their own metrics, IQMT offers the IQ XML schema of data representation, so this information can be easily imported. See an example in Table 4.

Filenames are important. They must follow this format:

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

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

- REFERENCE*-REFERENCE*

- SYSTEM*-REFERENCE*

```
<?xml version="1.0"?>
<!DOCTYPE iqmt SYSTEM "iqmt.dtd" []>
<IQREPORT metric="NEWMETRIC" ref="R2"
          score="0.6307" target="R0">
<S n="1">0.9960</S>
<S n="2">0.6250</S>
<S n="3">0.8519</S>
...
<S n="498">0.9985</S>
<S n="499">0.7129</S>
<S n="500">0.6408</S>
</IQREPORT>
```

Table 4: Example of XML IQREPORT representation file.

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 IQ) components. In the future we plan to add the KING and JACK components, which additionally require some more pairs:

- SYSTEM*-SYSTEM*
- REFERENCE*-SYSTEM*

For instance, in this case, 10 more XML files would become necessary:

- R0-S0.NEWMETRIC.xml
- R0-S1.NEWMETRIC.xml
- R1-S0.NEWMETRIC.xml
- R1-S1.NEWMETRIC.xml
- R2-S0.NEWMETRIC.xml
- R2-S1.NEWMETRIC.xml
- 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-REFERENCE’0**_...**REFERENCE’i**..._**REFERENCE’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 GTM-1 GTM-2 GTM-3
       MTR-exact MTR-stem MTR-wnstm MTR-wnsyn NIST-1
       NIST-2 NIST-3 NIST-4 NIST-5 RG-1 RG-2 RG-3
       RG-4 RG-L RG-SUs RG-Ss RG-W-1.2 NEWMETRIC
```

5. References

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

Banerjee, Satanjeev and Alon Lavie, 2005. 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*.

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

Giménez, Jesús, Enrique Amigó, and Chiori Hori, 2005. Machine translation evaluation inside qarla. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

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

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

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

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

Feedback

Discussion on this software as well as information about oncoming updates takes place on the IQMT google group, to which you can subscribe at:

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

and post messages at `IQMT@googlegroups.com`.