IQMT

A Framework for Automatic Machine Translation Evaluation based on Human Likeness

Technical Manual 2.0

Jesús Giménez
TALP Research Center, LSI Department
Universitat Politècnica de Catalunya
Jordi Girona Salgado 1–3. 08034, Barcelona
`jgimenez@lsi.upc.edu`

21th June 2007

Contents

Abstract

This report[1] presents a description and tutorial on the IQMT[2] package for Machine Translation Evaluation based on *'Human Likeness'*. IQMT intends to offer a common workbench on which MT evaluation metrics can be robustly utilized and combined for the purpose of MT system development. Current version includes a rich set of metrics operating at different linguistic levels (lexical, shallow syntactic, syntactic, and shallow semantic).

[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), and by the Spanish Ministry of Education and Science, projects OpenMT (TIN2006-15307-C03-02) and TRANGRAM (TIN2004-07925-C03-02).

[2] IQMT stands for Inside Qarla Machine Translation Evaluation Framework.

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 SVMTool 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 = ($path /home/me/IQMT-2.0/bin)
setenv PERL5LIB /home/me/IQMT-2.0/lib:$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`).
- Linguistic processors (under the ./tools directory) may require re-compilation:
  - SP metrics use the SVMTool (Giménez & Màrquez, 2004) (`http://www.lsi.upc.edu/~nlp/SVMT`).
  - DP metrics use the MINIPAR dependency parser (Lin, 1998) (`http://www.cs.ualberta.ca/~lindek/minipar.htm`).
  - CP metrics use the Charniak-Johnson Parser (Charniak & Johnson, 2005) (`ftp://ftp.cs.brown.edu/pub/nlparser/`).
  - NE metrics use the BIOS software (Surdeanu et al., 2005) (`http://www.lsi.upc.edu/~surdeanu/bios.html`).
  - SR metrics use the SwiRL software (Surdeanu & Turmo, 2005; Màrquez et al., 2005) (`http://www.lsi.upc.edu/~surdeanu/swirl.html`).

Getting all these software components to properly run may require a big initial effort. Most of them require in its turn several other smaller components. These may require again to set 'path' and PERL5LIB variables accordingly. For instance:

```
setenv PERL5LIB /home/me/IQMT-2.0/tools/METEOR.0.6:$PERL5LIB
setenv PATH /home/me/IQMT-2.0/tools/METEOR.0.6:$PATH
setenv PERL5LIB /home/me/IQMT-2.0/tools/PHRECO/lib:$PERL5LIB
setenv PERL5LIB /home/me/IQMT-2.0/tools/PHRECO/ml-2.3/lib:$PERL5LIB
setenv PATH /home/me/IQMT-2.0/tools/PHRECO/bin:$PATH
setenv PERL5LIB /home/me/IQMT-2.0/tools/SVMT/lib:$PERL5LIB
setenv PATH /home/me/IQMT-2.0/tools/SVMT/bin:$PATH
```

For those of you willing to complete the whole installation process, we hope the effort repays.

2 Introduction

Automatic evaluation metrics have notably accelerated the development cycle of Machine Translation (MT) systems in the last decade. Metrics play an essential role, allowing for fast numerical evaluations of translation quality on demand, which assist system developers in their everyday decisions. However, despite possible claims on the contrary, none of current metrics provides, in isolation, a *'global'* measure of quality. Indeed, all metrics focus on *'partial'* aspects.

As a result, a common methodological flaw in MT research is the mismatch between the system capabilities and the evaluation metrics used to measure these capabilities. Of course, the ultimate goal of system developers is to improve the overall quality of their MT system. But, in general, there is no magic recipe that allows us to improve all quality aspects at once. On the contrary, this goal is usually achieved in small steps. In each one of these steps (loops in the system development cycle), developers must identify and analyze possible sources of errors, focus on a specific type, think of a mechanism to solve them, implement it, and test it. Therefore, it is crucial for developers to count on 'appropriate' evaluation metrics; metrics which are able to capture possible improvements attained. Otherwise, they may be running the risk of too soon wrongly discarding fine mechanisms. Of course, at the same time, automatic evaluations should guarantee that partial improvements do not harm the overall system quality.

A large number of metrics, based on different assumptions and similarity criteria, have been suggested. However, there is a lack of a metric meta-evaluation framework which allows system developers to measure the suitability of metrics for a given translation scenario[3] in a fully-automatic and objective manner.

In order to satisfy this need, this report describes the IQMT[4] Framework for Machine Translation (meta-)evaluation, which is, to our knowledge, the first publicly available meta-evaluation software (Giménez & Amigó, 2006; Amigó et al., 2006). IQMT is based on the concept of 'Human Likeness' (Lin & Och, 2004b; Amigó et al., 2005; Kulesza & Shieber, 2004; Amigó et al., 2006). The underlying assumption is that human translations are better than automatic translations, and, therefore, a 'good' metric should never rank human translations lower

[3] At evaluation time, the translation scenario is determined by by the test bed, i.e., the sets of automatic and human reference translations.

[4] The IQMT 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`

(in quality) than automatic ones. Hence, inside IQ$_{\text{MT}}$ metrics are not evaluated in terms of correlation with human assessments (i.e., human acceptability, which is the current 'de facto' standard) but in terms of discriminative power, i.e., according to their ability to distinguish between automatic and human translations (i.e., human likeness). The main advantage of relying on human likeness is that human assessments are not required, and, therefore, meta-evaluation is objective, fully automatic, and updatable at no extra cost along time, as systems and metrics improve.

Additionally, IQ$_{\text{MT}}$ provides a mechanism to fight the 'metric bias' problem, by allowing system developers to work on metric combinations instead of on a single golden metric. Therefore, automatic metrics constitute a key ingredient inside IQ$_{\text{MT}}$. Current version includes a rich set of metrics operating at different linguistic levels (lexical, shallow syntactic, syntactic, and shallow semantic).

This material is intended to describe the ideas and methodology beneath the IQ$_{\text{MT}}$ framework as well as to serve as a tutorial for automatic 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 study on the evaluation of the Europarl Corpus Spanish-to-English translation task is presented in Section 6. Finally, further work is outlined in Section 7.

3 Fundamentals

IQ$_{\text{MT}}$ is based on QARLA (Amigó et al., 2005), a probabilistic framework originally designed for the evaluation of text summarization systems. 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 discriminative 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, $\mathrm{QUEEN}_{X,R}(a)$ 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:

$$\mathrm{KING}_{A,R}(X) = Prob(\forall a \in A :$$

$$\mathrm{QUEEN}_{X,R-\{r\}}(r) \geq \mathrm{QUEEN}_{X,R-\{r\}}(a)$$

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:

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

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 discriminative 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[5]. IQ*eval* provides an implementation of a simple algorithm which performs an approximate search in order to find a suboptimal set of metrics:

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

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.

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. IQMT 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*.

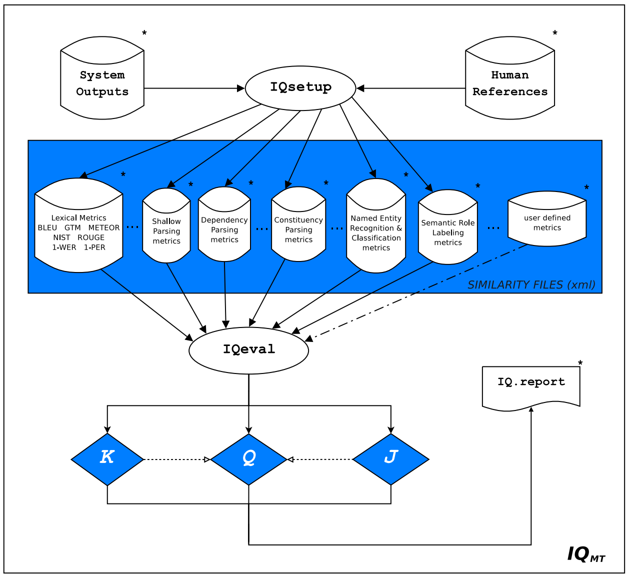

System Outputs
IQsetup
Human References
Lexical Metrics BLEU GTM METEOR NIST ROUGE 1-WER 1-PER
Shallow Parsing metrics
Dependency Parsing metrics
Constituency Parsing metrics
Named Entity Recognition & Classification metrics
Semantic Role Labeling metrics
user defined metrics
SIMILARITY FILES (xml)
IQeval
IQ.report
K
Q
J
IQMT

Figure 1: IQMT 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)
- IQMT 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[6] of IQ*setup* config file in Table 1.

You may then run IQ*setup*:

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

Options are:

-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.

[6] 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.

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

IQMT 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).

-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 by default).

-S `<set_name>` the set of system outputs to evaluate (all systems by default).

-M `<set_name>` the set of metrics (all metrics by default).

-T `<set_name>` the subset of sentences per system to evaluate (all sentences by default).

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

-TT enable trans-topic mode.

-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.

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.

```
[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*.

Specific sets 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 A Heterogeneous Metric Set

The set of similarity metrics is a dynamic component inside the IQMT 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 the lexical level, i.e., syntactic and semantic.

We have compiled a representative set of metrics at different linguistic levels. We have resorted to several existing metrics, and we have also developed new ones. Below, we group them according to the level at which they operate.

5.1 Lexical Similarity

IQMT 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[7]:

BLEU We use the default accumulated score up to the level of 4-grams (Papineni et al., 2001)[8].

NIST We use the default accumulated score up to the level of 5-grams (Doddington, 2002).

GTM We set to 1 the value of the *e* parameter (Melamed et al., 2003)[9].

METEOR We run all modules: 'exact', 'porter_stem', 'wn_stem' and 'wn_synonymy', in that order (Banerjee & Lavie, 2005)[10].

ROUGE We used the ROUGE-S* variant (skip bigrams with no max-gap-length). Stemming is enabled (Lin & Och, 2004a)[11].

Let us note that ROUGE and METEOR may consider stemming (i.e., morphological variations). Additionally, METEOR may perform a lookup for synonyms in WordNet (Fellbaum, 1998).

[7] WER and PER metrics have been also tested, but could not be released due to copyright reasons.

[8] We use mteval-kit-v10/mteval-v11b.pl for the computation of BLEU and NIST.

[9] We used GTM version 1.2.

[10] We used 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'.

5.2 Beyond Lexical Similarity

LinearB	On **Tuesday** several **missiles** and **mortar shells fell** in **southern Israel** , but there were **no casualties** .
Ref 1	Several **Qassam rockets** and **mortar shells** were **fired** on **southern Israel** today **Tuesday without victims** .
Ref 2	Several **Qassam rockets** and **mortars hit southern Israel** today **without causing any casualties** .
Ref 3	A number of **Qassam rockets** and **Howitzer missiles fell** over **southern Israel** today , **Tuesday**, **without causing any casualties** .
Ref 4	Several **Qassam rockets** and **mortar shells fell** today , **Tuesday** , on **southern Israel without causing any victim** .
Ref 5	Several **Qassam rockets** and **mortar shells fell** today , **Tuesday** , in **southern Israel without causing any casualties** .
Subject	Qassam rockets / Howitzer missiles / mortar shells
Action	fell / were fired / hit
Location	southern Israel
Time	Tuesday (today)
Result	no casualties / victims

Table 4: Case of Analysis (sentence #498 NIST 2005 Arabic-to-English Translation Exercise). Adequacy = 4, Fluency = 4, BLEU = 0.25.

It is an evidence that MT quality aspects are diverse. However, most current metrics, such as BLEU, limit their scope to the lexical dimension. This may result in 'unfair' evaluations. For instance, let us show in Table 4, a real case extracted from the NIST 2005 Arabic-to-English translation exercise, in which a high quality translation (by LinearB system) 'unfairly' attains a low score due to the low level of lexical matching. From all n-grams up to length four in the automatic translation only one 4-gram out of fifteen, two 3-grams out of sixteen, five 2-grams out of seventeen, and thirteen 1-grams out of eighteen can be found in at least one reference translation. Table 5 shows for these n-grams, in decreasing length ordering, the number of reference translations in which they co-occur.

The main problem with metrics based only on lexical similiarities, such as BLEU is that they are strongly dependent on the sublanguage represented by the set of human references available. In other words, their reliability depends on the heterogeneity (i.e., representativity) of the reference translations. These may in its turn depend not

n-gram	#occ	*n*-gram	#occ	*n*-gram	#occ
and mortar shells fell	2	casualties .	3	shells	3
and mortar shells	3	on	2	fell	3
mortar shells fell	2	Tuesday	4	southern	5
and mortar	3	several	4	Israel	5
mortar shells	3	missiles	1	,	3
shells fell	2	and	4	casualties	3
southern Israel	5	mortar	3	.	5

Table 5: Case of Analysis (sentence #498 NIST 2005 Arabic-to-English). Lexical matching.

only on the number of references, but on their lexica, grammar, style, etc.

The underlying problem is, in our opinion, that these metrics are too shallow, in the sense that, while the similarities between two sentences can take place at deeper linguistic levels, the limit their scope to the surface of lexical forms. Thus, they make an implicit use of linguistic knowledge, when, indeed, we believe that an explicit use of linguistic information could be very benefitial. Besides, current NLP technology allows for automatically obtaining such information. See, in Figure 2, an automatic syntactic/shallow-semantic representation (constituent trees, dependency relations, and semantic role labeling) for the automatic translation under study.

We argue that the degree of overlapping at more abstract levels is a far more robust indicator of actual MT quality. For instance, Figure 3 compares automatically obtained syntactic/shallow-semantic representations for the automatic translation in the previous example and one of the references. In first place, with respect to syntactic similarity, notice that a number of subtrees and dependencies are shared (particularly, noun phrases and prepositional phrases). Also notice that the main verbal form ('fell') is shared. As to the semantic roles associated, sentences share several arguments (A1, AM-TMP, and AM-LOC) with different degrees of lexical overlapping. All these features, that are making the difference in this case, are invisible to shallow metrics such as BLEU.

5.2.1 Linguistic Elements

Modeling linguistic features at levels further than the lexical level requires the usage of more complex linguistic structures. We have defined what we call *'linguistic elements'* (LEs). LEs are linguistic units, structures, or relationships, such that a sentence may be par-

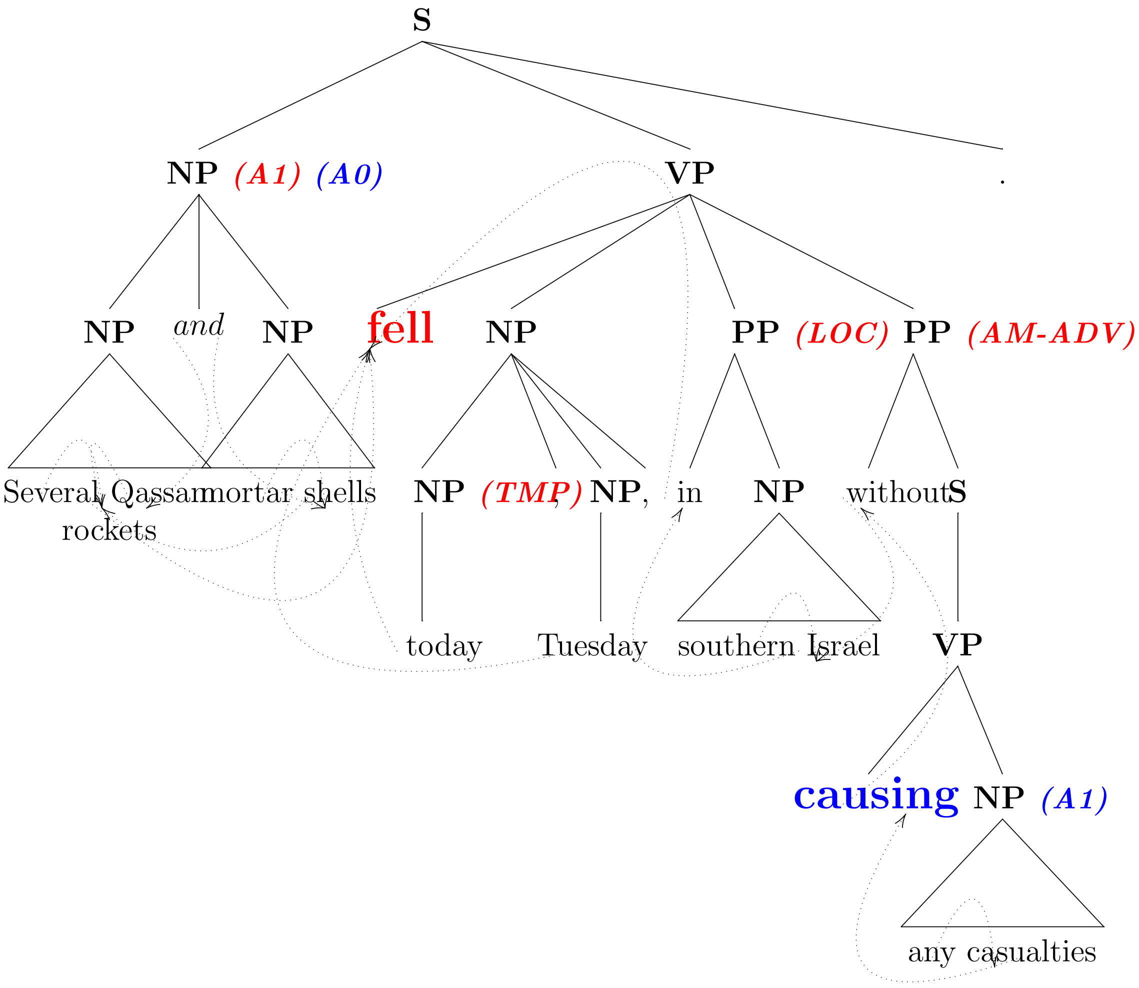
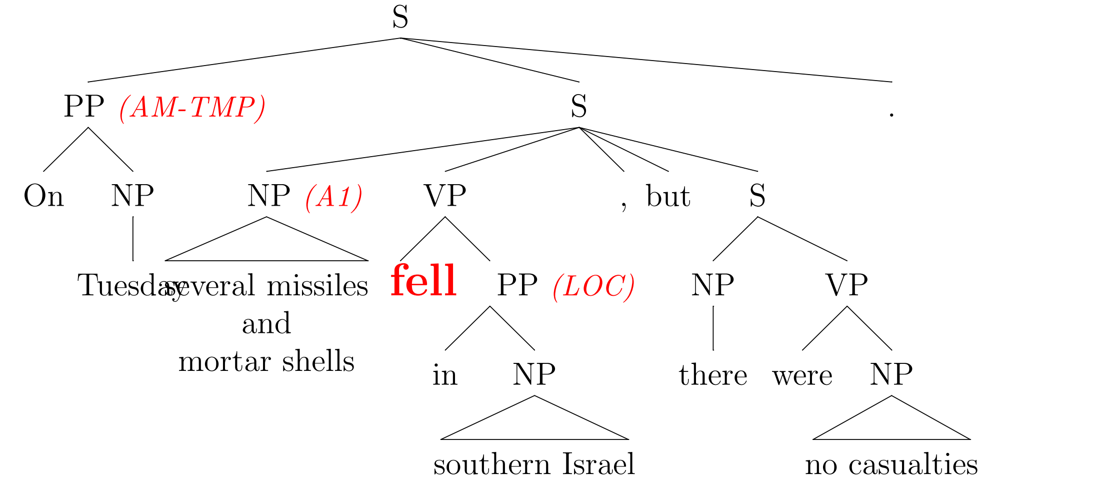


Figure 2: Linguistic Elements. A Syntactic/Shallow-Semantic Representation.



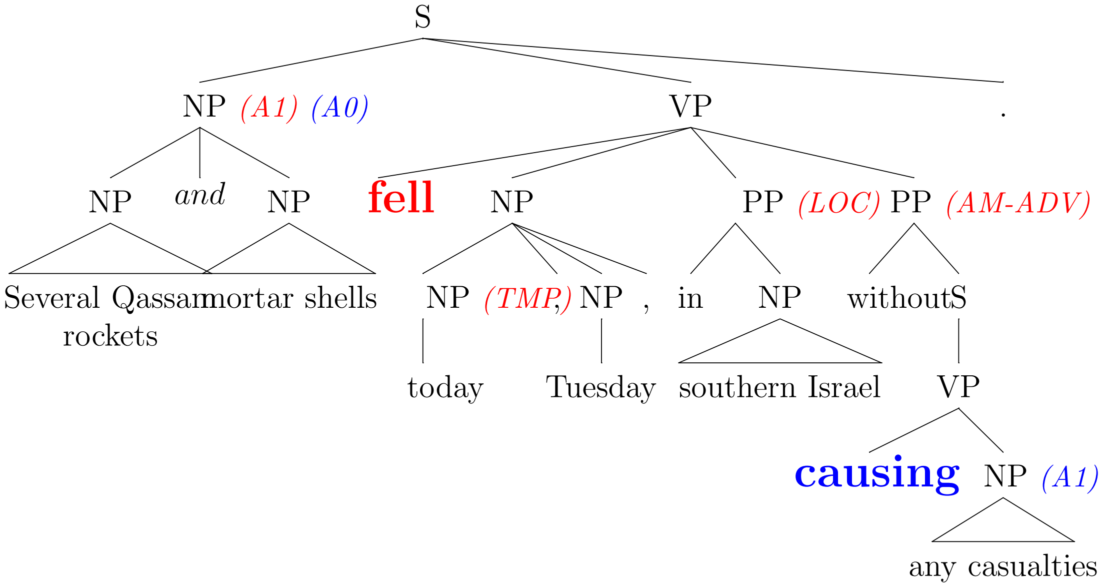


Figure 3: Case of Analysis (sentence #498). Syntactic/Shallow-Semantic Representation of LinearB System (top) against Human Reference #5 (bottom).

tially seen as a 'bag' of LEs. Possible kinds of LEs are: word forms, parts-of-speech, dependency relations, syntactic phrases, named entities, semantic roles, etc. Each LE may consist, in its turn, of one or more LEs, which we call 'items' inside the LE. For instance, a 'phrase' LE may consist of 'phrase' items, 'part-of-speech' (PoS) items, 'word form' items, etc. Items may be also combinations of LEs. For instance, a 'phrase' LE may be seen as a sequence of 'word-form:PoS' items.

In principle, LEs are not related to the 'basic elements' (BEs) defined by Hovy et al. (2006), used in the evaluation of automated summarization systems, although, in some way, BEs could be seen as a particular case of LEs.

5.2.2 Similarity Measures

We are interested in comparing linguistic structures, and linguistic units. LEs allow for comparisons at different granularity levels, and from different viewpoints. For instance, we might compare the semantic structure of two sentences (i.e., which actions, semantic arguments and adjuncts exist) or we might compare lexical units according to the semantic role they play inside the sentence. For that purpose, we use two very simple kinds of similarity measures over LEs: *'Overlapping'* and *'Matching'*. We provide a general definition:

Overlapping between items inside LEs, according to their type. Formally:

$$\text{Overlapping}(t) = \frac{\displaystyle\sum_{i \in (items_t(hyp)\ \cap\ items_t(ref))} count_{hyp}(i,t)}{\displaystyle\sum_{i \in (items_t(hyp)\ \cup\ items_t(ref))} max(count_{hyp}(i,t), count_{ref}(i,t))}$$

where t is the LE type[12], $items_t(s)$ refers to the set of items occurring inside LEs of type t in sentence s, and $count_s(i,t)$ denotes the number of times i appears in the sentence s inside a LE of type t. Thus, 'Overlapping' provides a rough measure of the proportion of items inside elements of a certain type which have been 'successfully' translated. We also introduce a coarser metric, **'Overlapping(*)'**, which considers average 'overlapping' over all types:

[12] LE types vary according to the specific LE class. For instance, in the case of Named Entities, types may be 'PER' (i.e., person), 'LOC' (i.e., location), 'ORG' (i.e., organization), etc.

$$\text{Overlapping}(\star) = \frac{\sum\limits_{t \in T} \sum\limits_{i \in (items_t(hyp)\ \cap\ items_t(ref))} count_{hyp}(i,t)}{\sum\limits_{t \in T} \sum\limits_{i \in (items_t(hyp)\ \cup\ items_t(ref))} max(count_{hyp}(i,t), count_{ref}(i,t))}$$

where T is the set of types.

Matching between items inside LEs, according to their type. Its definition is analogous to the 'Overlapping' definition, but in this case the relative order of the items is important. All items inside the same element are considered as a single unit (i.e., a sequence in left-to-right order). In other words, we are computing the proportion of 'fully' translated elements, according to their type. Formally:

$$\text{Matching}(t) = \frac{\sum\limits_{e \in (elems_t(hyp)\ \cap\ elems_t(ref))} count_{hyp}(e,t)}{\sum\limits_{e \in (elems_t(hyp)\ \cup\ elems_t(ref))} max(count_{hyp}(e,t), count_{ref}(e,t))}$$

where t is the LE type, $elems_t(s)$ refers to the set of LEs (as indivisible sequences of items) of type t in sentence s, and $count_s(e,t)$ denotes the number of times LE e of type t appears in the sentence s. We also introduce a coarser metric, **'Matching(*)'**, which considers average 'Matching' over all types.

$$\text{Matching}(\star) = \frac{\sum\limits_{t \in T} \sum\limits_{e \in (elems_t(hyp)\ \cap\ elems_t(ref))} count_{hyp}(e,t)}{\sum\limits_{t \in T} \sum\limits_{e \in (elems_t(hyp)\ \cup\ elems_t(ref))} max(count_{hyp}(e,t), count_{ref}(e,t))}$$

Notes:

- 'Overlapping' and 'Matching' operate on the assumption of a single reference translation. The reason is that, when it comes to more abstract levels, LEs inside the same sentence may be strongly interrelated, and therefore, similarities across reference translations may not be a reliable quality indicator. The extension to the multi-reference setting is computed by assigning the maximum value attained over all human references individually.
- 'Overlapping' and 'Matching' are general metrics. We may apply them to specific scenarios by defining the class of linguistic elements and items to be used. Below, we instantiate these measures over several particular cases.

- As to abbreviated nomenclature, the first two letters of metric names indicate the level of abstraction at which they operate. After that, we find the type of similarity computed. Overlapping and Matching measures are represented by the 'O' and 'M' symbols, respectively. Additionaly, these symbols may be accompanied by a subindex representing the type of LEs and items employed. For instance, 'SR-Or, w-*' operates at the level of semantic roles (SR), and represents average overlapping (O) between words (w) according to they role (r). If the LE and item types are not specified, it is assumed that the metric computes lexical overlapping over the top-level items available. For instance, these are all valid names for the previous metric: 'SR-Or, w-*', 'SR-Or-*', 'SR-Ow-*', and 'SR-O-*'.

5.3 Shallow Syntactic Similarity (SP)

Metrics based on shallow parsing (*'SP'*) analyze similarities at the level of PoS-tagging, lemmatization, and base phrase chunking. Outputs and references are automatically annotated using state-of-the-art tools. PoS-tagging and lemmatization are provided by the SVMT*ool*[13] package (Giménez & Màrquez, 2004), and base phrase chunking is provided by the P*hreco* software (Carreras et al., 2005). Tag sets for English are derived from the Penn Treebank (Marcus et al., 1993).

We instantiate 'Overlapping' over parts-of-speech and chunk types. The goal is to capture the proportion of lexical items correctly translated, according to their shallow syntactic realization:

SP-O_p-t Lexical overlapping according to the part-of-speech 't'. For instance, 'SP-O_p-NN' roughly reflects the proportion of correctly translated singular nouns. We also introduce a coarser metric, **'SP-O_p-*'** which computes average overlapping over all parts-of-speech.

SP-O_c-t Lexical overlapping according to the chunk type 't'. For instance, 'SP-O_c-NP' roughly reflects the successfully translated proportion of noun phrases. We also introduce a coarser metric, **'SP-O_c-*'** which considers the average overlapping over all chunk types.

SP-NIST(i)$_{iob}$-n Lexical overlapping over chunk IOB labels[14]. We also introduce a coarser metric, **'SP-O_{iob}-*'** which considers the average overlapping over all chunk types.

[13] http://www.lsi.upc.edu/~nlp/SVMTool

[14] IOB labels are used to denote the position (Inside, Outside, or Beginning of a chunk) and, if applicable, the type of chunk.

At a more abstract level, we use the NIST metric (Doddington, 2002) to compute accumulated/individual scores over sequences of:

Lemmas – $\mathbf{SP\text{-}NIST(i)}_{\boldsymbol{l}}\text{-}\boldsymbol{n}$
Parts-of-speech – $\mathbf{SP\text{-}NIST(i)}_{\boldsymbol{p}}\text{-}\boldsymbol{n}$
Base phrase chunks – $\mathbf{SP\text{-}NIST(i)}_{\boldsymbol{c}}\text{-}\boldsymbol{n}$

For instance, **'SP-NIST$_{l}$-5'** corresponds to the accumulated NIST score for lemma n-grams up to length 5, whereas 'SP-NISTi$_{p}$-5' corresponds to the individual NIST score for PoS 5-grams.

5.4 Syntactic Similarity

We have incorporated, with minor modifications, some of the syntactic metrics described by Liu and Gildea (2005) and Amigó et al. (2006) based on dependency and constituency parsing.

5.4.1 On Dependency Parsing (DP)

'DP' metrics capture similarities between dependency trees associated to automatic and reference translations. Dependency trees are provided by the MINIPAR[15] dependency parser (Lin, 1998) (a brief description of grammatical categories and relations may be found in Table 6 and Table 7).

Similarities are captured from different viewpoints:

DP-HWC(i)-*l* This metric corresponds to the HWC metric presented by Liu and Gildea (2005). All head-word chains are retrieved. The fraction of matching head-word chains of a given length, 'l', is computed. We have slightly modified this metric in order to distinguish three different variants according to the type of items head-word chains may consist of:

Lexical forms – $\mathbf{DP\text{-}HWC(i)}_{\boldsymbol{w}}\text{-}\boldsymbol{l}$

Grammatical categories – $\mathbf{DP\text{-}HWC(i)}_{\boldsymbol{c}}\text{-}\boldsymbol{l}$

Grammatical relations – $\mathbf{DP\text{-}HWC(i)}_{\boldsymbol{r}}\text{-}\boldsymbol{l}$

Average accumulated scores up to a given chain length may be used as well. For instance, 'DP-HWCi$_{w}$-4' retrieves the proportion of matching length-4 word-chains, whereas **'DP-HWC$_{w}$-4'** retrieves average accumulated proportion of matching word-chains up to length-4. Analogously, **'DP-HWC$_{c}$-4'**, and **'DP-HWC$_{r}$-4'** compute average accumulated proportion of category/relation chains up to length-4.

[15] http://www.cs.ualberta.ca/~lindek/minipar.htm

Type	**Description**
Det	Determiners
PreDet	Pre-determiners
PostDet	Post-determiners
NUM	numbers
C	Clauses
I	Inflectional Phrases
V	Verb and Verb Phrases
N	Noun and Noun Phrases
NN	noun-noun modifiers
P	Preposition and Preposition Phrases
PpSpec	Specifiers of Preposition Phrases
A	Adjective/Adverbs
Have	verb 'to have'
Aux	Auxilary verbs, e.g. should, will, does, ...
Be	Forms of verb 'to be': is, am, were, be, ...
COMP	Complementizer
VBE	'to be' used as a linking verb. E.g., I am hungry
V_N	verbs with one argument, i.e., intransitive verbs
V_N_N	verbs with two arguments, i.e., transitive verbs
V_N_I	verbs taking small clause as complement

Table 6: MINIPAR Grammatical categories.

DP-$\boldsymbol{O_l|O_c|O_r}$ These metrics correspond exactly to the LEVEL, GRAM and TREE metrics introduced by Amigó et al. (2006).

- **DP-$\boldsymbol{O_l}$-*l*** Overlapping between words hanging at level '*l*', or deeper.
- **DP-$\boldsymbol{O_c}$-*t*** Overlapping between words *directly hanging* from terminal nodes (i.e. grammatical categories) of type '*t*'.
- **DP-$\boldsymbol{O_r}$-*t*** Overlapping between words ruled by non-terminal nodes (i.e. grammatical relations) of type '*t*'.

Node types are determined by grammatical categories and relations defined by MINIPAR. For instance, 'DP-O_r-s' reflects lexical overlapping between subtrees of type 's' (subject). 'DP-O_c-A' reflects lexical overlapping between terminal nodes of type 'A' (Adjective/Adverbs). 'DP-O_l-4' reflects lexical overlapping between nodes hanging at level 4 or deeper. Additionally, we consider three coarser metrics (**'DP-$\boldsymbol{O_l}$-*'**, **'DP-$\boldsymbol{O_c}$-*'** and **'DP-$\boldsymbol{O_r}$-*'**) which correspond to the uniformly averaged values over all levels, categories, and relations, respectively.

Type	**Description**
appo	“ACME president, –appo-> P.W. Buckman”
aux	“should <- aux– resign”
be	“is <- be– sleeping”
by-subj	subject with passives
c	clausal complement “that <- c– John loves Mary”
cn	nominalized clause
comp1	first complement
desc	description
det	“the <- det ‘– hat”
gen	“Jane’s <- gen– uncle”
fc	finite complement
have	“have <- have– disappeared”
i	relation between a C clause and its I clause
inv-aux	inverted auxiliary: “Will <- inv-aux– you stop it?”
inv-be	inverted be: “Is <- inv-be– she sleeping”
inv-have	inverted have: “Have <- inv-have– you slept”
mod	relation between a word and its modifier
pnmod	post nominal modifier
p-spec	specifier of prepositional phrases
pcomp-c	clausal complement of prepositions
pcomp-n	nominal complement of prepositions
post	post determiner
pre	pre determiner
pred	predicate of a clause
rel	relative clause
obj	object of verbs
obj2	second object of ditransitive verbs
s	surface subject
sc	sentential complement
subj	subject of verbs
vrel	passive verb modifier of nouns
wh(a|n|p)	wh-elements at C-spec positions (a|n|p)

Table 7: MINIPAR Grammatical relations.

5.4.2 On Constituency Parsing (CP)

'CP' metrics capture similarities between constituency parse trees associated to automatic and reference translations. Constituency trees are provided by the Charniak-Johnson's Max-Ent reranking parser (Charniak & Johnson, 2005)[16].

CP-STM(i)-*l* This metric corresponds to the STM metric presented by Liu and Gildea (2005). All syntactic subpaths in the candidate and the reference trees are retrieved. The fraction of matching subpaths of a given length, 'l', is computed. For instance, 'CP-STMi-5' retrieves the proportion of length-5 matching subpaths. Average accumulated scores may be computed as well. For instance, **'CP-STM-9'** retrieves average accumulated proportion of matching subpaths up to length-9.

5.5 Shallow-Semantic Similarity

We have designed two new families of metrics, 'NE' and 'SR', which are intended to capture similarities over Named Entities (NEs) and Semantic Roles (SRs), respectively.

5.5.1 On Named Entities (NE)

'NE' metrics analyze similarities between automatic and reference translations by comparing the NEs which occur in them. Sentences are automatically annotated using the *BIOS*[17] package (Surdeanu et al., 2005). BIOS requires at the input shallow parsed text, which is obtained as described in Section 5.3. See the list of NE types in Table 8.

We define two types of metrics:

NE-O_e-*t* Lexical overlapping between NEs according to their type t. For instance, 'NE-O_e-PER' reflects lexical overlapping between NEs of type 'PER' (i.e., person), which provides a rough estimate of the successfully translated proportion of person names. The **'NE-O_e-*'** metric considers the average lexical overlapping over all NE types. This metric includes the NE type 'O' (i.e., Not-a-NE). We introduce another variant, **'NE-O_e-**'**, which considers only actual NEs.

NE-M_e-*t* Lexical matching between NEs according to their type t. For instance, 'NE-M_e-LOC' reflects the proportion of fully translated NEs of type 'LOC' (i.e., location). The **'NE-M_e-*'** metric

[16] ftp://ftp.cs.brown.edu/pub/nlparser/

[17] http://www.lsi.upc.edu/~surdeanu/bios.html

Type	**Description**
ORG	Organization
PER	Person
LOC	Location
MISC	Miscellaneous
O	Not-a-NE
DATE	Temporal expressions
NUM	Numerical expressions
ANGLE_QUANTITY	Quantities
DISTANCE_QUANTITY	
SIZE_QUANTITY	
SPEED_QUANTITY	
TEMPERATURE_QUANTITY	
WEIGHT_QUANTITY	
METHOD	Other
MONEY	
LANGUAGE	
PERCENT	
PROJECT	
SYSTEM	

Table 8: Named Entity types.

considers the average lexical matching over all NE types, this time excluding type 'O'.

Other authors have measured MT quality over NEs in the recent literature. In particular, the **'NE-M_e-*'** metric is similar to the **'NEE'** metric defined by Reeder et al. (2001).

5.5.2 On Semantic Roles (SR)

'SR' metrics analyze similarities between automatic and reference translations by comparing the SRs (i.e., arguments and adjuncts) which occur in them. Sentences are automatically annotated using the S*wi*RL[18] package (Surdeanu & Turmo, 2005; Màrquez et al., 2005). This package requires at the input shallow parsed text enriched with NEs, which is obtained as described in Section 5.5.1. See the list of SR types in Table 9.

We define three types of metrics:

SR-O_r-*t* Lexical overlapping between SRs according to their type *t*. For instance, 'SR-O_r-A0' reflects lexical overlapping between

[18] `http://www.lsi.upc.edu/~surdeanu/swirl.html`

Type	Description
A0 A1 A2 A3 A4 A5	arguments associated with a verb predicate, defined in the PropBank Frames scheme.
AA	Causative agent
AM-ADV	Adverbial (general-purpose) adjunct
AM-CAU	Causal adjunct
AM-DIR	Directional adjunct
AM-DIS	Discourse marker
AM-EXT	Extent adjunct
AM-LOC	Locative adjunct
AM-MNR	Manner adjunct
AM-MOD	Modal adjunct
AM-NEG	Negation marker
AM-PNC	Purpose and reason adjunct
AM-PRD	Predication adjunct
AM-REC	Reciprocal adjunct
AM-TMP	Temporal adjunct

Table 9: Semantic Roles.

'A0' arguments. **'SR-O_r-*'** considers the average lexical overlapping over all SR types.

SR-M_r-t Lexical matching between SRs according to their type t. For instance, the metric 'SR-M_r-AM-MOD' reflects the proportion of fully translated modal adjuncts. The **'SR-M_r-*'** metric considers the average lexical matching over all SR types.

SR-O_r This metric reflects 'role overlapping', i.e.. overlapping between semantic roles independently from their lexical realization.

Note that in the same sentence several verbs, with their respective SRs, may co-occur. However, the metrics described above do not distinguish between SRs associated to different verbs. In order to account for such a distinction we introduce a more restrictive version of these metrics ('SR-M_{rv}-t', 'SR-O_{rv}-t', **'SR-M_{rv}-*'**, **'SR-O_{rv}-*'**, and **'SR-O_{rv}'**), which require SRs to be associated to the same verb.

6 A case study: Europarl

In this section we present a case study on the application of the MT evaluation methodology proposed.

6.1 Experimental Setting

The ideal scenario for metric meta-evaluation should include 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 very similar. We have utilized the data from the *'Openlab 2006'* Initiative[19] promoted by the TC-STAR[20] Consortium. *'Openlab 2006'* data are entirely based on European Parliament Proceedings[21], covering April 1996 to May 2005.

We have focused 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++)[22].

SMT systems are built as described in (Giménez & Màrquez, 2005). As to 'SYSTRAN', we used the freely available on-line version[23]. 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 lexical metric variants (See Section 5.1).

6.2 Evaluating with Standard Metrics

First we analyze the individual behaviour of standard metrics outside QARLA. See results in Table 10. We use one representative from each

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

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

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

[22] This system is an improved version of the 'PB' system which uses information at the shallow-parsing level to build better translation models (Giménez & Màrquez, 2005).

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

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 10: MT quality according to several metrics outside QARLA.

family, the metric variant with highest KING value in the given test 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 IQMT

Inside the IQMT Framework systems are evaluated according to their 'Human Likeness'. Thus, we must trust the metric (or set of metrics) with highest discriminative power (highest KING), i.e. the metric which best identifies the features that distinguish between human translations and automatic translations. Table 11 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 discriminative 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}

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 11: Discriminative power of standard metrics (KING).

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 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 12: 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 12. 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 13 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 13: A case of error analysis, according to the QUEEN measure, in which the 'PB++' system outperforms the rest.

7 Further Work

We have successfully applied the proposed methodology to document-level evaluation. However, we have not applied the methodology to sentence-level evaluation which is crucial for error analysis.

Moreover, as we have seen, metrics based on linguistic information require automatic processors. This implies two important limitations. First, linguistic processors are not equally available for all languages. Second, usually they are too slow to allow for massive evaluations, as required, for instance, in the case of MT system development. In the future, we plan to incorporate more accurate, and possibly faster, linguistic processors, also for languages other than English, as they become publicly available.

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.

References

Amigó, E., Giménez, J., Gonzalo, J., & Màrquez, L. (2006). MT Evaluation: Human-Like vs. Human Acceptable. *Proceedings of COLING-ACL06.*

Amigó, E., Gonzalo, J., Peñas, A., & Verdejo, F. (2005). Qarla: a framework for the evaluation of automatic sumarization. *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics.*

Banerjee, S., & Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization.*

Carreras, X., Márquez, L., & Castro, J. (2005). Filtering-ranking perceptron learning for partial parsing. *Machine Learning*, *59*, 1–31.

Charniak, E., & Johnson, M. (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. *Proceedings of ACL.*

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. *Proceedings of the 2nd Internation Conference on Human Language Technology* (pp. 138–145).

Fellbaum, C. (Ed.). (1998). *WordNet. An Electronic Lexical Database.* The MIT Press.

Giménez, J., & Amigó, E. (2006). IQMT: A Framework for Automatic Machine Translation Evaluation. *Proceedings of the 5th LREC.*

Giménez, J., & Màrquez, L. (2004). SVMTool: A general POS tagger generator based on Support Vector Machines. *Proceedings of 4th LREC.*

Giménez, J., & Màrquez, L. (2005). Combining linguistic data views for phrase-based smt. *Proceedings of the Workshop on Building and Using Parallel Texts, ACL.*

Hovy, E., Lin, C.-Y., Zhou, L., , & Fukumoto, J. (2006). Automated Summarization Evaluation with Basic Elements. *Proceedings of the 5th LREC.*

Kulesza, A., & Shieber, S. M. (2004). A learning approach to improving sentence-level mt evaluation. *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation.*

Lin, C.-Y., & Och, F. J. (2004a). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statics. *Proceedings of ACL.*

Lin, C.-Y., & Och, F. J. (2004b). Orange: a method for evaluating automatic evaluation metrics for machine translation. *Proceedings of COLING.*

Lin, D. (1998). Dependency-based Evaluation of MINIPAR. *Proceedings of the Workshop on the Evaluation of Parsing Systems.*

Liu, D., & Gildea, D. (2005). Syntactic features for evaluation of machine translation. *Proceedings of ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization.*

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, *19*, 313–330.

Melamed, I. D., Green, R., & Turian, J. P. (2003). Precision and recall of machine translation. *Proceedings of HLT/NAACL.*

Màrquez, L., Surdeanu, M., Comas, P., & Turmo, J. (2005). Robust Combination Strategy for Semantic Role Labeling. *Proceedings of HLT/EMNLP.*

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). *Bleu: a method for automatic evaluation of machine translation, ibm research report, rc22176* (Technical Report). IBM T.J. Watson Research Center.

Reeder, F., Miller, K., Doyon, J., & White, J. (2001). The Naming of Things and the Confusion of Tongues: an MT Metric. *Proceedings of the Workshop on MT Evaluation "Who did what to whom?" at MT Summit VIII* (pp. 55–59).

Surdeanu, M., & Turmo, J. (2005). Semantic Role Labeling Using Complete Syntactic Analysis. *Proceedings of CoNLL Shared Task.*

Surdeanu, M., Turmo, J., & Comelles, E. (2005). Named Entity Recognition from Spontaneous Open-Domain Speech. *Proceedings of the 9th International Conference on Speech Communication and Technology (Interspeech).*