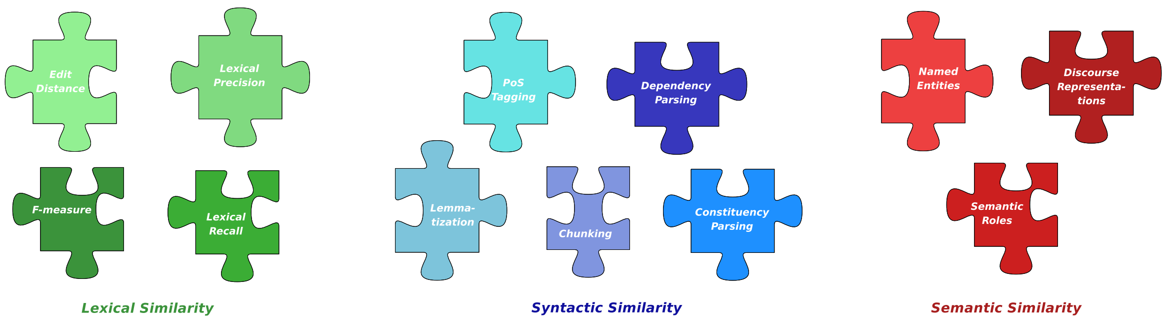
ULC



Technical Manual v0.4.2

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

September 2008

Abstract

This report[1] describes the ULC software for Automatic MT Evaluation, its fundamentals, installation and usage.

Based on the IQMT package, ULC evaluates translation quality through uniformly averaged linear combinations (i.e., arithmetic mean) of metric scores. Broadly speaking, ULC rewards those translations which are consistently among the top-scoring for all metrics. The most important difference with respect to standard evaluation methods is that ULC, just like IQMT, allows system developers to evaluate translation quality at different linguistic levels (lexical, syntactic and semantic). ULC has been successfully applied to several evaluation test beds, including the translation of European Parliament Proceedings from several European languages into English (Callison-Burch et al., 2008).

[1]The work reported has been funded by the Spanish Ministry of Education and Science, project OpenMT (TIN2006-15307-C03-02).

Contents

1 Installation

Download the ULC software[2] and unpack it by typing the following command:

```
tar xvfz ulc-0.4.2.tar.gz
```

This will generate a new folder named "ulc-0.4.2". To configure this module, cd to this directory 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
```

This will check whether all the required modules are installed or not. Prerequisites are:

- XML management:
  - XML::Twig 3.22[3]
  - XML::DOM 1.43 (requires, XML::Parser::PerlSAX, available inside libxml-perl-0.08)
  - XML::Parser 2.34 (requires expat)[4]
  - XML::RegExp 0.03
- Getopt::Long 2.37
- DB_File 1.814
- Data::Dumper 2.12
- strict 1.02
- IO 1.20
- IO::File 1.09
- POSIX 1.08
- Unicode::String 2.07
- File::ReadBackwards 1.04
- SVMTool::SVMTAGGER 1.3 (available inside SVMTool v1.3)[5]

If there is some module missing, most likely you will find it in the "./perl-cpan/" and "./soft/" directories. Otherwise, check out the CPAN repository[6].

Then, build the package by typing:

```
make
```

If you have write access to the installation directories, you may then install it so it is available to all other users:

```
make install
```

[2] http://www.lsi.upc.edu/~jgimenez/ulc-0.4.2.tar.gz
[3] http://www.xmltwig.com/xmltwig/
[4] http://sourceforge.net/projects/expat/
[5] http://www.lsi.upc.edu/~nlp/SVMT
[6] http://search.cpan.org/

Otherwise, remember to properly set the PERL5LIB variable so Perl programs may find ULC modules:

```
export PERL5LIB=$PERL5LIB:/home/me/ulc-0.4.2/lib
```

In any case, the "./tools" directory must be included in the PERL5LIB variable:

```
export PERL5LIB=$PERL5LIB:/home/me/ulc-0.4.2/tools/
```

Finally, include the folder containing ULC executable files in the PATH variable:

```
export PATH=$PATH:/home/me/ulc-0.4.2/bin
```

1.1 External Components

ULC relies on several external components for metric computation. All are located in the "./tools" directory, and some may require re-compilation. In this case, simply 'cd' to the corresponding directory and follow the instructions in the 'README' file.

1.1.1 Borrowing Metrics

- METEOR (in the METEORrequires the Lingua::Stem::Snowball module and WordNet[7]. In its turn, WordNet requires Tcl/tk[8]. After installation, you must properly set the WNHOME and PATH variables:

  ```
  export PATH=$PATH:/usr/local/WordNet-3.0/bin
  export WNHOME=/usr/local/WordNet-3.0
  ```

- GTM requires Java[9].

1.1.2 Borrowing Linguistic Processors

- SP metrics use the SVMTool (Giménez & Màrquez, 2004a)[10], which requires Perl.
- CP metrics use the Charniak-Johnson Parser (Charniak & Johnson, 2005) (`ftp://ftp.cs.brown.edu/pub/nlparser/`), which requires C++.
- DP metrics use the MINIPAR dependency parser (Lin, 1998)[11]. MINIPAR requires the GNU Standard C++ Library v3 (libstdc++5).
- NE metrics use the BIOS software (Surdeanu et al., 2005)[12].
- SR metrics use the SwiRL software (Surdeanu & Turmo, 2005; Màrquez et al., 2005)[13], which requires JAVA and C++, as well as BIOS, and the Charniak-Johnson Parser.

[7] `http://wordnet.princeton.edu`
[8] `http://www.tcl.tk/`
[9] `http://www.java.com`
[10] `http://www.lsi.upc.edu/~nlp/SVMT`
[11] `http://www.cs.ualberta.ca/~lindek/minipar.htm`
[12] `http://www.lsi.upc.edu/~surdeanu/bios.html`
[13] `http://www.lsi.upc.edu/~surdeanu/swirl.html`

- DR metrics use the C&C Tools[14], which requires C++ and SWI PROLOG[15]. Detailed installation instructions are available in the C&C Tools website[16]. Remember to install the BOXER component. BOXER expects the prolog interpreter under the name of 'pl'. Thus, you may need to edit the PROLOG variable in the Makefile. Alternatively, you can create a soft link:

  ```
  ln -s /usr/bin/swipl /usr/bin/pl
  ```

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:

```
#METEOR
export PERL5LIB=$PERL5LIB:/home/jgimenez/soft/ulc-0.4.2/tools/METEOR.0.6
export PATH=$PATH:/home/jgimenez/soft/ulc-0.4.2/tools/METEOR.0.6
#SVMTool
export PERL5LIB=$PERL5LIB:/home/jgimenez/soft/ulc-0.4.2/tools/svmtool-1.3/lib
export PATH=$PATH:/home/jgimenez/soft/ulc-0.4.2/tools/svmtool-1.3/bin
#MINIPAR
export MINIPATH=/home/jgimenez/soft/ulc-0.4.2/tools/minipar/data
export PATH=$PATH:/home/jgimenez/soft/ulc-0.4.2/tools/minipar/pdemo
```

2 Usage

2.1 Parameters

If you run the ULC application with no parameters you will get the following output:

```
[user@machine ulc-0.4.2]$ ./bin/ULC

Usage: ULC  [options]  <ULC.config>

Options:

  - m <metrics>          : set of metrics (setname according to ULC.config)
  - s <systems>          : set of systems (setname according to ULC.config)
  - r <references>       : set of references (setname according to ULC.config)
  - remake               : remake metric computations
  - v                    : verbosity
  - version              : version number
  - help                 : this help

Example: ULC -v ULC.config
```

ULC requires a config file (ULC.config, see an example in the './sample/empty/' directory). This file is intended to be modified by the user for:

[14] http://svn.ask.it.usyd.edu.au/trac/candc/
[15] http://www.swi-prolog.org/
[16] http://svn.ask.it.usyd.edu.au/trac/candc/wiki/Installation

1. Specifying:
    - path to ULC software (e.g., "PATH=/home/me/ulc-0.4.2")
    - target language (e.g., "LANG=ENG") (only ENGlish supported in this version)
    - output case (all|lc) (e.g., "CASE=all")
        - all: upper and lower mixed case (default)
        - lc: lower case
    - input format (xml|raw) (e.g., "INPUT=xml") (NIST 'xml' by default)
    - system output files (e.g, "sys=./data/system01.sgm")
    - reference files (e.g., "ref=./data/reference01.sgm")
2. Defining:
    - metric sets (e.g., "2_metrics_LEX= RG-W-1.2 MTR-wnsyn"). By default, a heuristically pre-defined set of metrics containing several representatives from each linguistic level is used.

      M_h = { RG-W-1.2, MTR-wnsyn, CP-STM-4, DP-HWC-c-4, DP-HWC-r-4, DP-Or-*, SR-Or-*-b, SR-Mr-*-b, SR-Or-b, DR-Or-*-b, DR-Orp-*-b }

      See the definition of the "metrics_DEFAULT" metric set in the ULC.config file. This metric set has exhibited high correlation with human assessments over several test beds, including the translation of European Parliament Proceedings from several European languages into English (Callison-Burch et al., 2008). A complete list of metrics available may be found in the "metrics_all" set defined in the ULC.config file.
    - system sets (e.g., "2_first_systems=S0 S1"). By default, all systems are evaluated.
    - reference sets (e.g., "2_first_refs=R0 R1"). By default, all references are used.

2.2 Output

ULC complies with instructions given in the Metrics MATR 2008 evaluation plan v1.1[17], both for input and output formats. Accordingly, each run of the ULC meta-metric generates a series of ".src" files, which are stored in the current working directory. For instance, in the case of the sample test bed:

```
[user@machine ulc-0.4.2/sample/gold]$ ls -1

./data
./S0
./S1
...
./S7
./system01-doc.scr
./system01-seg.scr
./system01-sys.scr
./system02-doc.scr
./system02-seg.scr
./system02-sys.scr
```

[17] http://www.nist.gov/speech/tests/metricsmatr/2008/doc/mm08_evalplan_v1.1.pdf

```
...
./system08-doc.scr
./system08-seg.scr
./system08-sys.scr
...
ULC.config
ULC.log
```

2.3 By-pass products

ULC generates, as a by-pass product, several intermediate files:

- metric computation intermediate files are stored in the "./tmp" folder inside the working directory. This folder will end up empty after all successful executions. Upon abnormal exit, the content of this folder could be used for debugging.
- linguistic analysis files, stored in the data directory.
- metric similarities, stored in a separate folder for each system output (e.g., "./SO", "./S1", ..., "./S7"), and grouped according to the reference set employed (e.g., "./S0/R0_R1_R2_R3/").

This allows for "dramatically" speeding up the following runs of the ULC meta-metric.

2.4 Testing ULC

Test 1

1. 'cd' to the "./sample/" directory.
2. Type:

   ```
   cp -r empty test1
   ```

3. 'cd' to the "./sample/test1/ directory.
4. edit the "ULC.config" file so the PATH variable points to the ULC source directory:

   ```
   #-- path to ULC
   PATH=/home/me/ulc-0.4.2
   ```

5. Type:

   ```
   ULC -v ULC.config >& ULC.log &
   ```

   This will compute the ULC meta-metric over a small sample test bed, based on the default set of metrics. This process will allow you to check that the ULC software and all external components work correctly.

   Execution will take several minutes, since ULC performs a deep linguistic analysis of both automatic translations and human references.

6. When the process ends, you may compare results and by-pass intermediate files with those in the "./sample/gold" directory.

Test 2

1. 'cd' to the "./sample/" directory.
2. Type:

   ```
   cp -r test1 test2
   ```

3. 'cd' to the "./sample/test2/ directory.
4. Type:

   ```
   ULC -v -m metrics_LEX -s 2\_first\_systems -r ref\_0 ULC.config
   ```

   This will compute ULC over "ROUGE-w-1.2" and "METEOR-wnsyn" (i.e., ULC(x) = (ROUGE-w-1.2(x) + METEOR-wnsyn(x)) / 2, for each segment x), for the two first systems (declared in top-down order), against the first reference (again following top-down order).
5. Check results as compared to those in the '../test1' directory.

2.5 Running ULC at the Metrics MATR Challenge

Create a proper configuration file[18]. Then, run ULC over the following different metric sets in order. After each run, remember to move score files (i.e., *.scr) to a different directory. Otherwise, they will be overwritten.

ULC$_h$

Run ULC over the heuristically predefined set of metrics[19]:

```
ULC -v -m metrics_DEFAULT ULC.config
```

which is equivalent to:

```
ULC -v ULC.config
```

This first run will take several hours. The following runs should only take some minutes (linguistic processors are no longer necessary), or even seconds (most metrics have been computed as a by-pass product).

ULC

Run ULC over the set of metrics of optimal adequacy over the 'mt06' development test bed provided[20]:

```
ULC -v -m metrics_OPT_mt06_adequacy ULC.config
```

[18]Use the 'ULC.config' file located in the './sample/empty' directory, and copy the metric set definitions in it.

[19]M_h = { RG-W-1.2, MTR-wnsyn, CP-STM-4, DP-HWC-c-4, DP-HWC-r-4, DP-Or-*, SR-Or-*-b, SR-Mr-*-b, SR-Or-b, DR-Or-*-b, DR-Orp-*-b }

[20]$M_{opt.mt06.adequacy}$ = { RG-L, RG-W-1.2, MTR-wnsyn, Ol, DP-Or-*, DR-Orp-*-i }

DP

Try dependency overlapping[21]:

```
ULC -v -m metrics_DP ULC.config
```

SR

Try semantic role overlapping[22]:

```
ULC -v -m metrics_SR ULC.config
```

DR

Finally, try two different variants of overlapping over discourse representations. First, lexical overlapping[23]:

```
ULC -v -m metrics_DR ULC.config
```

Then, morphosyntactic overlapping[24]:

```
ULC -v -m metrics_DRp ULC.config
```

2.6 A Note on Efficiency

ULC relies on a rich set of linguistic metrics (see Section 4), which in their turn rely on automatic linguistic processors. Thus, the efficiency of ULC ultimately depends on the efficiency of linguistic processors. For instance, running ULC over the 'Metrics Matr' development set, based on the heuristically pre-defined set of metrics described above, takes around 4 hours on an Intel(R) Core(TM)2 CPU 2.13GHz machine with a 2GB RAM.

3 Uniformly-averaged Linear Combinations of Metric Scores

Integrating the scores conferred by different metrics into a single measure seems the most natural and direct way to improve over the individual quality of current metrics. This solution requires two important ingredients:

Combination Strategy, i.e., how to combine several metric scores into a single score. We distinguish between *parametric* and *non-parametric* approaches. In parametric approaches the contribution of each metric to the global score is individually weighted through an associated parameter. In contrast, in the non-parametric case, metric contribution is based on a global non-parameterized criterion.

Meta-Evaluation Criterion, i.e., how to evaluate the quality of a metric combination. There exist at least two different meta-evaluation criteria: human likeness (i.e., the metric ability to discern between automatic and human translations) and human acceptability (i.e., correlation with human assessments).

[21] M_{dp} = { DP-Or-* }
[22] M_{sr} = { SR-Or-*-b }
[23] M_{dr} = { DR-Or-*-b }
[24] M_{drp} = { DR-Orp-*-b }

ULC *emulates* a non-parametric scheme based on human acceptability by working with uniformly averaged linear combinations of metric scores. Our approach is similar to that of Liu and Gildea (2007) except that in our case all the metrics in the combination are equally important[25]. In other words, ULC is indeed a particular case of a parametric scheme, in which the contribution of each metric is not adjusted. Formally:

$$\mathrm{ULC}_X(a, R) = \frac{1}{|X|} \sum_{x \in X} x(a, R)$$

where X is the metric set, and $x(a, R)$ is the similarity between the automatic translation a and the set of references R, for the given test case, according to the metric x.

We evaluate metric quality in terms of correlation with human assessments at the sentence level (R_{snt}).

4 A Heterogeneous Set of Metrics

For our study, we have compiled a rich set of metric variants at 5 different linguistic levels (lexical, shallow-syntactic, syntactic, shallow-semantic and semantic). We have resorted to several existing metrics, and we have also developed new ones[26]. Although from different viewpoints, and based on different similarity assumptions, in all cases, translation quality is measured by comparing automatic translations against a set of human reference translations. In the following subsections, we provide a description of the metrics according to the linguistic level at which they operate.

4.1 Lexical Similarity

We have included several variants from different standard metrics (e.g., BLEU, NIST, GTM, METEOR, ROUGE, WER PER and TER)[27]. Below we list all the variants included in our study:

- **BLEU-*n* | BLEUi-*n*:** Accumulated and individual BLEU scores for several n-gram levels ($n = 1...4$) (Papineni et al., 2001). We use version '11b' of the NIST MT evaluation kit[28] for the computation of BLEU scores. Seven variants are computed[29].
- **NIST-*n* | NISTi-*n*:** Accumulated and individual NIST scores for several n-gram levels ($n = 1...5$) (Doddington, 2002). We use version '11b' of the NIST MT evaluation kit for the computation of NIST scores. Nine variants are computed[30].
- **GTM-*e*:** General Text Matching F-measure (Melamed et al., 2003). We use GTM version 1.4. Three variants, corresponding to different values of the e parameter controlling the reward for longer matchings ($e \in \{1, 2, 3\}$), are computed.
- **METEOR:** We use METEOR version 0.6. (Banerjee & Lavie, 2005). Four variants are computed[31]:
  - $\mathbf{METEOR_{exact}}$ → running 'exact' module.

[25] That would be assuming that all metrics operate in the same range of values, which is not always the case.
[26] Current version, available only for English being the target language, includes a rich set of more than 500 metrics.
[27] The list of the variants selected is also available in Table 6.
[28] The NIST MT evaluation kit is available at `http://www.nist.gov/speech/tests/mt/scoring/`.
[29] We use 'BLEU' to refer to the 'BLEU-4' variant. 'BLEU-1' and 'BLEUi-1' refer to the same metric variant.
[30] We use 'NIST' to refer to the 'NIST-5' variant. 'NIST-1' and 'NISTi-1' refer to the same metric variant.
[31] We use 'METEOR' to refer to the '$\mathrm{METEOR_{wnsyn}}$' variant.

- $\textbf{METEOR}_{\textbf{stem}}$ → running 'exact' and 'porter_stem' modules, in that order. This variant considers morphological variations through the Porter stemmer (Porter, 2001).
- $\textbf{METEOR}_{\textbf{wnstm}}$ → running 'exact', 'porter_stem' and 'wn_stem' modules, in that order. This variant includes morphological variations obtained through WordNet (Fellbaum, 1998).
- $\textbf{METEOR}_{\textbf{wnsyn}}$ → running 'exact', 'porter_stem', 'wn_stem' and 'wn_synonymy' modules, in that order. This variant performs a lookup for synonyms in WordNet.

- **ROUGE:** We use ROUGE version 1.5.5 (Lin & Och, 2004). We consider morphological variations through stemming. Options are '-z SPL -2 -1 -U -m -r 1000 -n 4 -w 1.2 -c 95 -d'. Eight variants are computed:
  - **ROUGE-*n*** → for several n-gram lengths ($n = 1...4$)
  - $\textbf{ROUGE}_{\boldsymbol{L}}$ → longest common subsequence (LCS).
  - $\textbf{ROUGE}_{\boldsymbol{S\star}}$ → skip bigrams with no max-gap-length.
  - $\textbf{ROUGE}_{\boldsymbol{SU\star}}$ → skip bigrams with no max-gap-length, including unigrams.
  - $\textbf{ROUGE}_{\boldsymbol{W}}$ → weighted longest common subsequence (WLCS) with weighting factor $w = 1.2$.
- **WER:** Word Error Rate. We use $1 - \text{WER}$ (Nießen et al., 2000).
- **PER:** Position-independent Word Error Rate. We use $1 - \text{PER}$ (Tillmann et al., 1997).
- **TER:** Translation Edit Rate. We use $1 - \text{TER}$ (Snover et al., 2006).

4.2 Beyond Lexical Similarity

MT quality aspects are diverse. However, metric families listed in Section 4.1 limit their scope to the lexical dimension. This may result, in unfair evaluations. For instance, let us show in Table 1, a real case extracted from the NIST 2005 Arabic-to-English translation exercise[32]. A high quality translation (by LinearB system) according to human assessments (adequacy = 4 / 5, fluency = 4 / 5) unfairly attains a low BLEU score (BLEU = 0.25). This is 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 2 shows, for these n-grams in decreasing length order, the number of reference translations in which they occur.

The main problem with metrics based only on lexical similarities 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 only on the number of references, but on their lexica, grammar, style, etc. Besides, while similarities between two sentences can take place at deeper linguistic levels, lexical metrics limit their scope to the surface. We believe that an explicit use of linguistic information could be very beneficial. Besides, current NLP technology allows for automatically obtaining such information.

Thus, we argue that the degree of overlapping at more abstract levels is a far more robust indicator of actual MT quality. For instance, Figure 1 compares automatically obtained syntactico-semantic representations for the automatic translation in the previous example (top) and reference #5 (bottom)[33]. In first place, with respect to syntactic similarity, notice that a number of subtrees

[32] The case corresponds to sentence 498 in the test set.

[33] Part-of-speech and syntactic notation are based on the Penn Treebank (Marcus et al., 1993). Notation for semantic roles is based on the Proposition Bank (Palmer et al., 2005). We distinguish semantic roles associated to different verbs by indexing them with the position the related verb would occupy in a left-to-right list of verbs, starting at position 1.

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 1: NIST 2005 Arabic-to-English. A Case of Analysis (sentence #498)

***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 2: NIST 2005 Arabic-to-English. A Case of Analysis (sentence #498). Lexical matching

are shared (particularly, noun phrases and prepositional phrases). Also notice that the main verbal form ('fell') is shared. As to the semantic roles associated, predicates in both 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.

4.2.1 Linguistic Elements

Modeling linguistic features at deeper linguistic levels requires the usage of more complex linguistic structures. We will refer to linguistic units, structures, or relationships as *linguistic elements* (LEs). Possible kinds of LEs could be, for instance, word forms, parts of speech, dependency relations, syntactic constituents, named entities, semantic roles, discourse representations, etc. A sentence, thus, may be seen as a bag of LEs. Each LE may consist, in its turn, of one or more LEs, which we call items inside the LE. For instance, a phrase constituent LE may consist of part-of-speech items, word form items, etc. LEs may also consist of combinations of items. For instance, a phrase constituent LE may be seen as a sequence of 'word-form:part-of-speech' items.

Hovy et al. (2006) defined a similar type of linguistic structures, so-called basic elements (BEs), for the evaluation of automated summarization systems. Their method consisted in breaking down

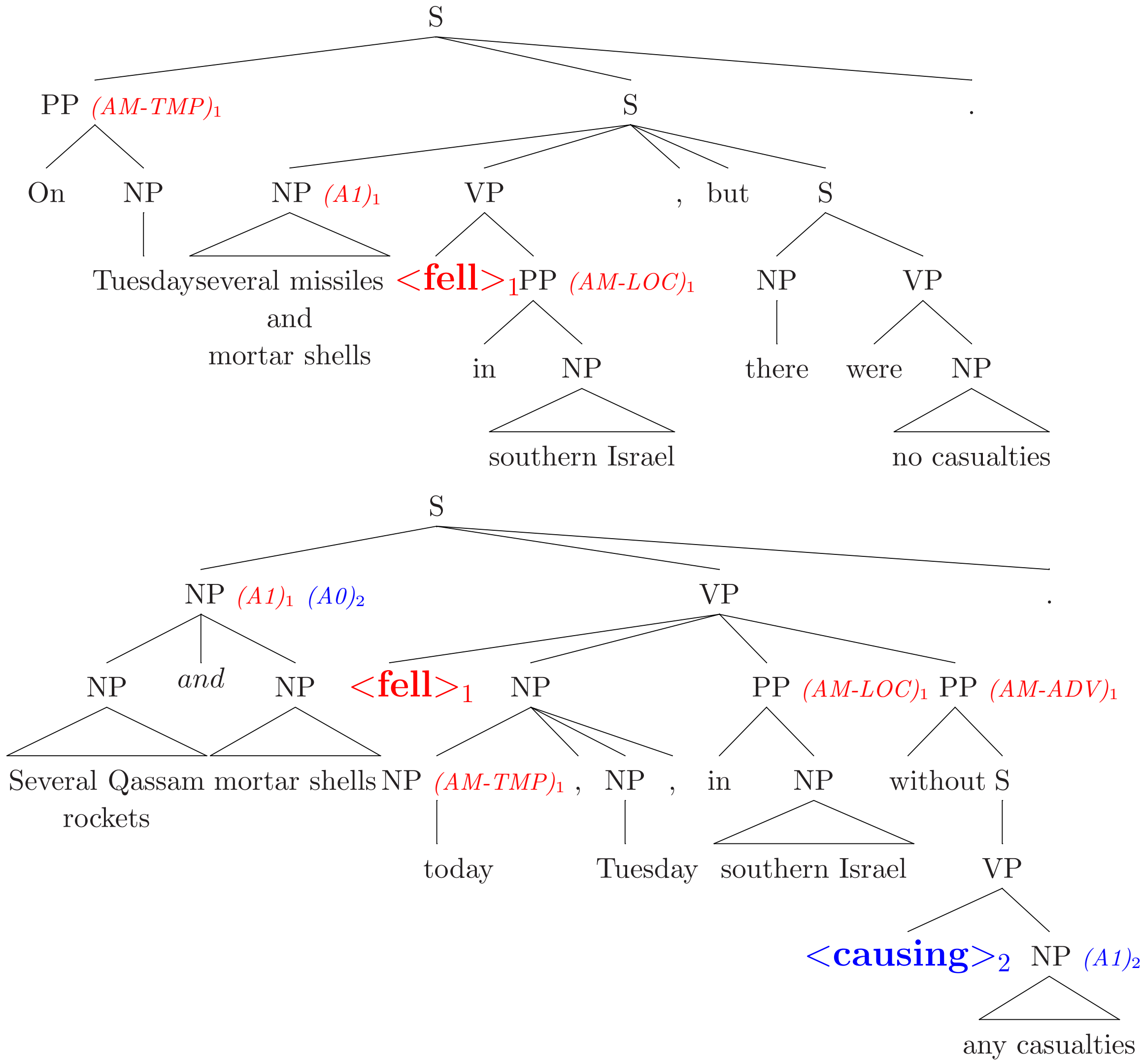


Figure 1: NIST 2005 Arabic-to-English. A Case of Analysis (sentence #498). Syntactico-semantic Representation

reference sentences into sets of BEs before comparing system outputs against them. However, in contrast to LEs, they limited the information captured by BEs to the syntactic level, whereas LEs allow for representing any kind of linguistic information. Thus, BEs could be actually seen as a particular case of LEs.

4.2.2 Similarity Measures over Linguistic Elements

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 syntactic/semantic structure of two sentences (e.g., which verbs, semantic arguments and adjuncts exist) or we might compare lexical units according to the syntactic/semantic role they play inside the sentence. We use two very simple kinds of similarity measures over LEs: *Overlapping* and *Matching*. Below, we provide general definitions which will be instantiated over particular cases in the following subsections:

- **Overlapping** *between items inside LEs, according to their type.* Overlapping provides a rough measure of the proportion of items inside elements of a certain type that have been successfully translated. Formally:

$$\text{Overlapping}(t) = \frac{\displaystyle\sum_{i \in (\text{items}_t(\text{hyp}) \,\cap\, \text{items}_t(\text{ref}))} \text{count}_{\text{hyp}}(i,t)}{\displaystyle\sum_{i \in (\text{items}_t(\text{hyp}) \,\cup\, \text{items}_t(\text{ref}))} \max(\text{count}_{\text{hyp}}(i,t), \text{count}_{\text{ref}}(i,t))}$$

  where t is the LE type, 'hyp' and 'ref' refer, respectively, to the candidate and reference translations, $\text{items}_t(s)$ refers to the set of items occurring inside LEs of type t in sentence s, and $\text{count}_s(i,t)$ denotes the number of times i appears in sentence s inside a LE of type t. LE types vary according to the specific LE class. For instance, in the case of the 'named entity' class, types may be 'PER' (i.e., person), 'LOC' (i.e., location), 'ORG' (i.e., organization), etc. In the case of the 'semantic role' class, types may be 'A0' (i.e., prototypical subject), 'AM-TMP' (i.e., temporal adjunct), 'AM-MNR' (i.e., manner adjunct), etc.

  We also introduce a coarser metric, Overlapping($\star$), which considers the averaged overlapping over all types:

$$\text{Overlapping}(\star) = \frac{\displaystyle\sum_{t \in T} \sum_{i \in (\text{items}_t(\text{hyp}) \,\cap\, \text{items}_t(\text{ref}))} \text{count}_{\text{hyp}}(i,t)}{\displaystyle\sum_{t \in T} \sum_{i \in (\text{items}_t(\text{hyp}) \,\cup\, \text{items}_t(\text{ref}))} \max(\text{count}_{\text{hyp}}(i,t), \text{count}_{\text{ref}}(i,t))}$$

  where T is the set of all LE types associated to the given LE class. For instance, we may define a metric which computes average lexical overlapping over all semantic roles types. This would roughly estimate to what degree translated lexical items play the expected semantic role in the context of the full candidate sentence.

- **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:

$$\text{Matching}(t) = \frac{\displaystyle\sum_{e \in (elems_t(\text{hyp}) \,\cap\, elems_t(\text{ref}))} \text{count}_{\text{hyp}}(e,t)}{\displaystyle\sum_{e \in (elems_t(\text{hyp}) \,\cup\, elems_t(\text{ref}))} \max(\text{count}_{\text{hyp}}(e,t), \text{count}_{\text{ref}}(e,t))}$$

where t is the LE type, and $elems_t(s)$ refers to the set of LEs (as indivisible sequences of consecutive items) of type t in sentence s.

As in the case of 'Overlapping', we introduce a coarser metric, Matching($\star$), which considers the averaged matching over all types:

$$\text{Matching}(\star) = \frac{\displaystyle\sum_{t \in T} \sum_{e \in (elems_t(\text{hyp}) \,\cap\, elems_t(\text{ref}))} \text{count}_{\text{hyp}}(e,t)}{\displaystyle\sum_{t \in T} \sum_{e \in (elems_t(\text{hyp}) \,\cup\, elems_t(\text{ref}))} \max(\text{count}_{\text{hyp}}(e,t), \text{count}_{\text{ref}}(e,t))}$$

4.2.3 Notes on Overlapping/Matching Measures

1. 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.
2. 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. In subsections 4.3 to 4.6.1, these measures are instantiated over several particular cases.
3. As to abbreviated nomenclature, the first two letters of metric names identify the LE class, which indicates the level of abstraction at which they operate. In this document, we use 'SP' for shallow parsing, 'DP' for dependency parsing, 'CP' for constituency parsing, 'NE' for named entities, 'SR' for semantic roles, and 'DR' for discourse representations. Then, we find the type of similarity computed. Overlapping and Matching measures are represented by the 'O' and 'M' symbols, respectively. Additionally, these symbols may be accompanied by a subindex representing the type of LEs and items employed. For instance, 'SR-O_{rl}-$\star$' operates at the level of semantic roles (SR), and represents average <u>O</u>verlapping among <u>l</u>exical items according to their <u>r</u>ole. 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 also valid names for the 'SR-O_{rl}-$\star$' metric: 'SR-O_r-$\star$', 'SR-O_l-$\star$', and 'SR-O-$\star$'. In the following sections and chapters, we use 'SR-O_r-$\star$' equivalent, and similarly for other metrics and LE classes.

4.2.4 Lexical Overlapping

We instantiate the overlapping measure at the lexical level, by defining the 'O_l' metric, which computes lexical overlapping directly over word forms. As an example, Table 3 shows the computation of the 'O_l' score for the case depicted in Figure 1, as compared to lexical precision, recall and F-measure. A and H denote, respectively, the automatic translation and the human reference. Text

A	on **tuesday several** missiles **and mortar shells fell in southern israel ,** but there were no **casualties .**
H	**several** qassam rockets **and mortar shells fell** today , **tuesday , in southern israel** without causing any **casualties .**

$A \cap H = \{$ 'tuesday', 'several', 'and', 'mortar', 'shells', 'fell', 'in', 'southern', 'israel', ',', 'casualties', '.' $\}$

$A \cup H = \{$ 'on', 'tuesday', 'several', 'missiles', 'and', 'mortar', 'shells', 'fell', 'in', 'southern', 'israel', ',', 'but', 'there', 'were', 'no', 'casualties', '.', 'qassam', 'rockets', 'today', ',', 'without', 'causing', 'any' $\}$

$O_l = \frac{|A| \cap |H|}{|A| \cup |H|} = \frac{12}{25}$ $\quad P = \frac{|A| \cap |H|}{|A|} = \frac{12}{18}$ $\quad R = \frac{|A| \cap |H|}{|H|} = \frac{12}{19}$ $\quad F = \frac{2 * P * R}{P + R} = \frac{2 * \frac{12}{18} * \frac{12}{19}}{\frac{12}{18} + \frac{12}{19}}$

Table 3: Lexical overlapping score for the case from Table 1

has been lower cased. It can be observed that lexical overlapping is, indeed, just another simple method for balancing precision and recall.

4.2.5 An Example Beyond the Lexical Level

Table 4 shows an example on how to compute average lexical overlapping among semantic roles, i.e., SR-O_r-($\star$), for the case depicted in Figure 1. The semantic role labeler detected one argument ('A1_1') and two adjuncts ('AM-TMP_1' and 'AM-LOC_1') in the automatic translation, whereas three arguments ('$\text{A1})_1$', 'A0_2', and 'A1_2') and three adjuncts ('AM-TMP_1', 'AM-LOC_1' and 'AM-ADV_1') were detected for the human reference. Associated LE representations are showed for each LE type. We also provide individual lexical overlapping scores, and average overlapping.

4.3 Shallow Syntactic Similarity

Metrics based on shallow parsing (*SP*) analyze similarities at the level of parts of speech (PoS), word lemmas, and base phrase chunks. Sentences are automatically annotated using the SVMTool (Giménez & Màrquez, 2004b), Freeling (Carreras et al., 2004) and Phreco (Carreras et al., 2005) linguistic processors, as described in Appendix B, Section B.1. 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. Two metrics have been defined:

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, whereas 'SP-O_p-VBN' reflects the proportion of correctly translated past participles. We also define the 'SP-O_p-$\star$' metric, which computes the average lexical overlapping over all parts of speech.

SP-O_c-*t* Lexical overlapping according to the base phrase chunk type '*t*'. For instance, 'SP-O_c-NP', and 'SP-O_c-VP' respectively reflect the successfully translated proportion of noun and verb phrases. We also define the 'SP-O_c-$\star$' metric, which computes the average lexical overlapping over all chunk types.

$\mathrm{A}_{\mathrm{A1}} = \{$ **'several'**, 'missiles', **'and'**, **'mortar'**, **'shells'** $\}$
$\mathrm{H}_{\mathrm{A1}} = \{$ 'several', 'qassam', 'rockets, 'and', 'mortar', 'shells', 'any', 'casualties' $\}$

$\mathrm{A}_{\mathrm{A0}} = \emptyset$
$\mathrm{H}_{\mathrm{A0}} = \{$ 'several', 'qassam', 'rockets, 'and', 'mortar', 'shells' $\}$
$\mathrm{A}_{\text{AM-TMP}} = \{$ 'on', 'tuesday' $\}$
$\mathrm{H}_{\text{AM-TMP}} = \{$ 'today' $\}$
$\mathrm{A}_{\text{AM-LOC}} = \{$ **'in'**, **'southern'**, **'israel'** $\}$
$\mathrm{H}_{\text{AM-LOC}} = \{$ 'in', 'southern', 'israel' $\}$
$\mathrm{A}_{\text{AM-ADV}} = \emptyset$
$\mathrm{H}_{\text{AM-ADV}} = \{$ 'without', 'causing', 'any', 'casualties' $\}$

$\text{SR-}O_r(\text{A1}) = \frac{4}{9}$
$\text{SR-}O_r(\text{A0}) = \frac{0}{6}$
$\text{SR-}O_r(\text{AM-TMP}) = \frac{0}{3}$
$\text{SR-}O_r(\text{AM-LOC}) = \frac{3}{3}$
$\text{SR-}O_r(\text{AM-ADV}) = \frac{0}{4}$

$\text{SR-}O_r(\star) = \frac{4+0+0+3+0}{9+6+3+3+4} = \frac{7}{25}$

Table 4: Average semantic role (lexical) overlapping score for the case from Table 1

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

SP-NIST(i)-*n* Lemmas.

SP-NIST(i)$_{\boldsymbol{p}}$-*n* Parts of speech.

SP-NIST(i)$_{\boldsymbol{c}}$-*n* Base phrase chunks.

SP-NIST(i)$_{\boldsymbol{iob}}$-*n* Chunk IOB labels[34].

For instance, 'SP-NIST$_{\boldsymbol{l}}$-5' corresponds to the accumulated NIST score for lemma n-grams up to length 5, whereas 'SP-NISTi$_{\boldsymbol{p}}$-5' corresponds to the individual NIST score for PoS 5-grams. 'SP-NIST$_{\boldsymbol{iob}}$-2' corresponds to the accumulated NIST score for IOB n-grams up to length 2, whereas 'SP-NISTi$_{\boldsymbol{c}}$-4' corresponds to the individual NIST score for chunk 4-grams. A complete list of SP metric variants is available in Appendix A, Table 7.

4.4 Syntactic Similarity

4.4.1 On Dependency Parsing (DP)

DP metrics capture similarities between dependency trees associated to automatic and reference translations. Dependency trees are obtained using the MINIPAR parser (Lin, 1998), as described in Appendix B, Section B.2. We use two types of metrics:

DP-$\boldsymbol{O_l|O_c|O_r}$ These metrics compute lexical overlapping between dependency trees from three different viewpoints:

- **DP-$\boldsymbol{O_l}$-*l*** Overlapping between words hanging at the same level, $l \in [1..9]$, or deeper. For instance, 'DP-O_l-4' reflects lexical overlapping between nodes hanging at level 4 or deeper. Additionally, we define the 'DP-O_l-$\star$' metric, which corresponds to the averaged values over all levels.
- **DP-$\boldsymbol{O_c}$-*t*** Overlapping between words *directly hanging* from terminal nodes (i.e., grammatical categories) of type 't'. For instance, 'DP-O_c-A' reflects lexical overlapping between terminal nodes of type 'A' (Adjective/Adverbs). Additionally, we define the 'DP-O_c-$\star$' metric, which corresponds to the averaged values over all categories.
- **DP-$\boldsymbol{O_r}$-*t*** Overlapping between words ruled by non-terminal nodes (i.e., grammatical relations) of type 't'. For instance, 'DP-O_r-s' reflects lexical overlapping between subtrees of type 's' (subject). Additionally, we define the 'DP-O_r-$\star$' metric, which corresponds to the averaged values over all relation types.

DP-HWC(i)-*l* This metric corresponds to the Head-Word Chain Matching (HWCM) metric presented by Liu and Gildea (2005). All head-word chains are retrieved. The fraction of matching head-word chains of a given length, $l \in [1..9]$, between the candidate and the reference translation is computed. Average accumulated scores up to a given chain length may be used as well. Opposite to the formulation by Liu and Gildea, in our case reference translations are considered individually. Moreover, we define three variants of this metric according to the items head-word chains may consist of:

- **DP-HWC(i)$_{\boldsymbol{w}}$-*l*** chains consist of words.
- **DP-HWC(i)$_{\boldsymbol{c}}$-*l*** chains consist of grammatical categories, i.e., parts of speech.
- **DP-HWC(i)$_{\boldsymbol{r}}$-*l*** chains consist of grammatical relations.

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

For instance, 'DP-HWCi$_{\boldsymbol{w}}$-4' retrieves the proportion of matching word-chains of length-4, whereas 'DP-HWC$_{\boldsymbol{w}}$-4' retrieves average accumulated proportion of matching word-chains *up to* length-4. Analogously, 'DP-HWC$_{\boldsymbol{c}}$-4', and 'DP-HWC$_{\boldsymbol{r}}$-4' compute average accumulated proportion of category/relation chains up to length-4.

The extension of 'DP-HWC' metrics to the multi-reference setting is computed by assigning to each metric the maximum value attained when individually comparing to all the trees associated to the different human references.

A complete list of DP metric variants is available in Appendix A, Table 8.

4.4.2 On Constituency Parsing (CP)

CP metrics analyze similarities between constituency parse trees associated to automatic and reference translations. Constituency trees are obtained using the Charniak-Johnson's Max-Ent reranking parser (Charniak & Johnson, 2005), as described in Appendix B, Section B.2. Three types of metrics are defined:

CP-STM(i)-*l* This metric corresponds to the Syntactic Tree Matching (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, $l \in [1..9]$, is computed. Average accumulated scores up to a given tree depth d may be used as well. 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.

The extension of the 'CP-STM' metrics to the multi-reference setting is computed by assigning to each metric the maximum value attained when individually comparing to all the trees associated to the different human references.

CP-$\boldsymbol{O_p}$-*t* Similarly to the 'SP-O_p' metric, this metric computes lexical overlapping according to the part-of-speech 't'.

CP-$\boldsymbol{O_c}$-*t* These metrics compute lexical overlapping according to the phrase constituent type 't'. The difference between these metrics and 'SP-O_c-t' variants is in the phrase scope. In contrast to base phrase chunks, constituents allow for phrase embedding and overlapping.

We also define the 'CP-O_p-$\star$' and 'CP-O_c-$\star$' metrics, which compute the average lexical overlapping over all parts of speech and phrase constituents, respectively.

A complete list of CP metric variants is available in Appendix A, Table 9.

4.5 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.

4.5.1 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 package (Surdeanu et al., 2005), as described in Appendix B, Section B.3. BIOS requires at the input shallow parsed text, which is obtained as described in Section 4.3. At the output, BIOS returns the text enriched with NE information. We have defined 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. We also define the 'NE-O_e-$\star$' metric, which considers the average lexical overlapping over all NE types. Note that this metric considers only actual NEs, i.e., it excludes the NE type 'O' (Not-a-NE). Thus, this metric is useless when no NEs appear in the translation. In order to improve its recall, we introduce the 'NE-O_e-$\star\star$' variant, which , considers overlapping among all items, including those of type 'O'.

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-$\star$' metric considers the average lexical matching over all NE types, excluding type 'O'.

A complete list of NE metric variants is available in Appendix A, Table 10.

4.5.2 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 the predicates. Sentences are automatically annotated using the SwiRL package (Surdeanu & Turmo, 2005), as described in Appendix B, Section B.3. This package requires at the input shallow parsed text enriched with NEs, which is obtained as described in Section 4.5.1. At the output, SwiRL returns the text annotated with SRs following the notation of the Proposition Bank (Palmer et al., 2005). We have defined 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 'A0' arguments. We also consider 'SR-$\boldsymbol{O_r}$-$\star$', which computes 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. Again, 'SR-M_r-$\star$' 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 verb predicates, with their respective argument structures, 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-$\boldsymbol{M_{rv}}$-$\boldsymbol{t}$', 'SR-$\boldsymbol{O_{rv}}$-$\boldsymbol{t}$', 'SR-$\boldsymbol{M_{rv}}$-$\star$', 'SR-O_{rv}-$\star$', and 'SR-O_{rv}'), which require SRs to be associated to the same verb.

A complete list of SR metric variants is available in Appendix A, Table 11.

4.6 Semantic Similarity

4.6.1 On Discourse Representations (DR)

At the properly semantic level, we have developed a novel family of metrics based on the Discourse Representation Theory (DRT) by Kamp (1981). DRT is a theoretical framework offering a representation language for the examination of contextually dependent meaning in discourse. A discourse is represented in a discourse representation structure (DRS), which is essentially a variation of first-order predicate calculus —its forms are pairs of first-order formulae and the free variables that occur in them. *DR* metrics analyze similarities between automatic and reference translations by comparing their respective DRSs. Sentences are automatically analyzed using the

C&C Tools (Clark & Curran, 2004), as described in Appendix B, Section B.4. DRS are viewed as semantic trees. As an example, Table 5 shows the DRS for *"Every man loves Mary."*.

```
drs([[4]:Y],
    [[4]:named(Y,mary,per,0),
     [1]:imp(drs([[1]:X],
                 [[2]:pred(X,man,n,1)]),
             drs([[3]:E],
                 [[3]:pred(E,love,v,0),
                  [3]:rel(E,X,agent,0),
                  [3]:rel(E,Y,patient,0)]))])
```

```
 _________          ____________________   ________________
| x0      |        | x1                 | | x2             |
|_________|        |____________________| |________________|
| man(x0) | ==> (| named(x1,mary,per) |A| love(x2)       |)
|_________|        |____________________| | event(x2)      |
                                          | agent(x2,x0)   |
                                          | patient(x2,x1) |
                                          |________________|
```

Table 5: An example of DRS-based semantic tree

We have defined three groups of metrics over DRSs:

DR-STM(i)-*l* This metric is similar to the *'STM'* metric defined by Liu and Gildea (2005), in this case applied to DRSs instead of constituent trees. All semantic subpaths in the candidate and the reference trees are retrieved. The fraction of matching subpaths of a given length, $l \in [1..9]$, is computed. Average accumulated scores up to a given tree depth d may be used as well. For instance, 'DR-STMi-5' retrieves the proportion of length-5 matching subpaths. Average accumulated scores may be computed as well. For instance, 'DR-STM-9' retrieves average accumulated proportion of matching subpaths up to length-9.

DR-O_r-*t* These metrics compute lexical overlapping between discourse representation structures (i.e., discourse referents and discourse conditions) according to their type 't'. For instance, 'DR-O_r-pred' roughly reflects lexical overlapping between the referents associated to predicates (i.e., one-place properties), whereas 'DR-O_r-imp' reflects lexical overlapping between referents associated to implication conditions. We also introduce a the 'DR-O_r-$\star$' metric, which computes average lexical overlapping over all DRS types.

DR-O_{rp}-*t* These metrics compute morphosyntactic overlapping (i.e., between grammatical categories –parts-of-speech– associated to lexical items) between discourse representation structures of the same type. We also define the 'DR-O_{rp}-$\star$' metric, which computes average morphosyntactic overlapping over all DRS types.

Note that in the case of some complex conditions, such as implication or question, the respective order of the associated referents in the tree is important. We take this aspect into account by making the order information explicit in the construction of the semantic tree. We also make explicit the type, symbol, value and date of conditions which have type, symbol, value or date, such as predicates, relations, named entities, time expressions, cardinal expressions, or anaphoric conditions.

A complete list of DR metric variants is available in Appendix A, Table 12.

4.7 Improved Sentence Level Behavior

By inspecting particular cases we have found that in many cases metrics are unable to produce any evaluation result. The number of unscored sentences is particularly significant in the case of SR metrics. Several reasons explain this fact. The most important is that metrics based on deep linguistic analysis rely on automatic processors trained on out-of-domain data, which are, thus, prone to error.

A natural and direct solution, in order to improve their performance, could be to back off to a measure of lexical similarity in those cases in which linguistic processors are unable to produce any linguistic analysis. This should significantly increase their recall. With that purpose, we have designed two new variants for each of these metrics. Given a linguistic metric x, we define:

- $\boldsymbol{x_b}$ $\longrightarrow$ by backing off to lexical overlapping, O_l, only when the linguistic processor was not able to produce a parsing. Lexical scores are conveniently scaled so that they are in a similar range to x scores. Specifically, we multiply them by the average x score attained over all other test cases for which the parser succeeded. Formally, given a test case t belonging to a set of test cases T:

$$x_b(t) = \begin{cases} O_l(t) * \frac{\sum_{j \in ok(T)} x(j)}{|ok(T)|} & \text{if parsing}(t) \text{ failed} \\ x(t) & \text{otherwise} \end{cases}$$

  where $ok(T)$ is the subset of test cases in T which were successfully parsed.

- $\boldsymbol{x_i}$ $\longrightarrow$ by linearly interpolating x and O_l scores for all test cases, via arithmetic mean:

$$x_i(t) = \frac{x(t) + O_l(t)}{2}$$

In both cases, system-level scores are calculated by averaging over all sentence-level scores.

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

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 MT and/or Summarization.*

Bos, J. (2005). Towards Wide-Coverage Semantic Interpretation. *Proceedings of the Sixth International Workshop on Computational Semantics (IWCS-6)* (pp. 42–53).

Bos, J., Clark, S., Steedman, M., Curran, J. R., & Hockenmaier, J. (2004). Wide-Coverage Semantic Representations from a CCG Parser. *Proceedings of the 20th International Conference on Computational Linguistics (COLING)* (pp. 1240–1246).

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., & Schroeder, J. (2008). Further meta-evaluation of machine translation. *Proceedings of the Third Workshop on Statistical Machine Translation* (pp. 70–106). Columbus, Ohio: Association for Computational Linguistics.

Carreras, X., Chao, I., Padró, L., & Padró, M. (2004). FreeLing: An Open-Source Suite of Language Analyzers. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)* (pp. 239–242).

Carreras, X., Màrquez, 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 the 43rd Annual Meeting of the Association for Computational Linguistics (ACL).*

Clark, S., & Curran, J. R. (2004). Parsing the WSJ using CCG and Log-Linear Models. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 104–111).

Collins, M. (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. *Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1–8).

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. *Proceedings of the 2nd International Conference on Human Language Technology* (pp. 138–145).

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

Giménez, J., & Màrquez, L. (2004a). Fast and Accurate Part-of-Speech Tagging: The SVM Approach Revisited. *Recent Advances in Natural Language Processing III* (pp. 153–162). Amsterdam: John Benjamin Publishers. ISBN 90-272-4774-9.

Giménez, J., & Màrquez, L. (2004b). SVMTool: A general POS tagger generator based on Support Vector Machines. *Proceedings of 4th International Conference on Language Resources and Evaluation (LREC)* (pp. 43–46).

Hovy, E., Lin, C.-Y., Zhou, L., & Fukumoto, J. (2006). Automated Summarization Evaluation with Basic Elements. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)* (pp. 899–902).

Kamp, H. (1981). A Theory of Truth and Semantic Representation. *Formal Methods in the Study of Language* (pp. 277–322). Amsterdam: Mathematisch Centrum.

Lin, C.-Y., & Och, F. J. (2004). Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statics. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL).*

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 MT and/or Summarization* (pp. 25–32).

Liu, D., & Gildea, D. (2007). Source-Language Features and Maximum Correlation Training for Machine Translation Evaluation. *Proceedings of the 2007 Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)* (pp. 41–48).

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330.

Màrquez, L., Surdeanu, M., Comas, P., & Turmo, J. (2005). Robust Combination Strategy for Semantic Role Labeling. *Proceedings of the Joint Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP)*.

Melamed, I. D., Green, R., & Turian, J. P. (2003). Precision and Recall of Machine Translation. *Proceedings of the Joint Conference on Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*.

Nießen, S., Och, F. J., Leusch, G., & Ney, H. (2000). An Evaluation Tool for Machine Translation: Fast Evaluation for MT Research. *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, *31*(1), 71–106.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2001). *Bleu: a method for automatic evaluation of machine translation, RC22176* (Technical Report). IBM T.J. Watson Research Center.

Porter, M. (2001). The Porter Stemming Algorithm.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., & Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)* (pp. 223–231).

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)*.

Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., & Sawaf, H. (1997). Accelerated DP based Search for Statistical Translation. *Proceedings of European Conference on Speech Communication and Technology*.

Toutanova, K., Klein, D., & Manning, C. D. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of Joint Conference on Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)* (pp. 173–180).

A Metric Sets

1-WER	= {	1-WER }
1-PER	= {	1-PER }
1-TER	= {	1-TER }
BLEU	= {	BLEU-1, BLEU-2, BLEU-3, BLEU-4, BLEUi-2, BLEUi-3, BLEUi-4 }
GTM	= {	GTM-1, GTM-2, GTM-3 }
METEOR	= {	METEOR_{exact}, METEOR_{stem}, METEOR_{wnstm}, METEOR_{wnsyn} }
NIST	= {	NIST-1, NIST-2, NIST-3, NIST-4, NIST-5, NISTi-2, NISTi-3, NISTi-4, NISTi-5 }
ROUGE	= {	ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE_{L}, $\text{ROUGE}_{S\star}$, $\text{ROUGE}_{SU\star}$, ROUGE_{W} }
LEX	= {	1-PER, 1-WER, 1-TER, BLEU-1, BLEU-2, BLEU-3, BLEU-4, BLEUi-2, BLEUi-3, BLEUi-4, GTM-1, GTM-2, GTM-3, NIST-1, NIST-2, NIST-3, NIST-4, NIST-5, NISTi-2, NISTi-3, NISTi-4, NISTi-5, ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, ROUGE_{L}, $\text{ROUGE}_{S\star}$, $\text{ROUGE}_{SU\star}$, ROUGE_{W}, METEOR_{exact}, METEOR_{stem}, METEOR_{wnstm}, METEOR_{wnsyn} }

Table 6: Metrics at the Lexical Level

SP = { SP-NIST$_c$-1, SP-NIST$_c$-2, SP-NIST$_c$-3, SP-NIST$_c$-4, SP-NIST$_c$-5,
SP-NISTi$_c$-2, SP-NISTi$_c$-3, SP-NISTi$_c$-4, SP-NISTi$_c$-5, SP-NIST$_{iob}$-1,
SP-NIST$_{iob}$-2, SP-NIST$_{iob}$-3, SP-NIST$_{iob}$-4, SP-NIST$_{iob}$-5, SP-NISTi$_{iob}$-2,
SP-NISTi$_{iob}$-3, SP-NISTi$_{iob}$-4, SP-NISTi$_{iob}$-5, SP-NIST$_l$-1, SP-NIST$_l$-2,
SP-NIST$_l$-3, SP-NIST$_l$-4, SP-NIST$_l$-5, SP-NISTi$_l$-2, SP-NISTi$_l$-3,
SP-NISTi$_l$-4, SP-NISTi$_l$-5, SP-O_c-$\star$, SP-O_c-ADJP, SP-O_c-ADVP,
SP-O_c-CONJP, SP-O_c-INTJ, SP-O_c-LST,
SP-O_c-NP, SP-O_c-O, SP-O_c-PP, SP-O_c-PRT,
SP-O_c-SBAR, SP-O_c-UCP, SP-O_c-VP, SP-O_p-#,
SP-O_p-$, SP-$O_p$-”, SP-$O_p$-(, SP-$O_p$-), SP-$O_p$-$\star$,
SP-O_p-, , SP-O_p-., SP-O_p-:, SP-O_p-CC, SP-O_p-CD,
SP-O_p-DT, SP-O_p-EX, SP-O_p-F, SP-O_p-FW, SP-O_p-IN,
SP-O_p-J, SP-O_p-JJ, SP-O_p-JJR, SP-O_p-JJS, SP-O_p-LS,
SP-O_p-MD, SP-O_p-N, SP-O_p-NN, SP-O_p-NNP,
SP-O_p-NNPS, SP-O_p-NNS, SP-O_p-P, SP-O_p-PDT,
SP-O_p-POS, SP-O_p-PRP, SP-O_p-PRP$, SP-$O_p$-R,
SP-O_p-RB, SP-O_p-RBR, SP-O_p-RBS, SP-O_p-RP,
SP-O_p-SYM, SP-O_p-TO, SP-O_p-UH, SP-O_p-V,
SP-O_p-VB, SP-O_p-VBD, SP-O_p-VBG, SP-O_p-VBN,
SP-O_p-VBP, SP-O_p-VBZ, SP-O_p-W, SP-O_p-WDT,
SP-O_p-WP, SP-O_p-WP$, SP-$O_p$-WRB, SP-$O_p$-“,
SP-NIST$_p$-1, SP-NIST$_p$-2, SP-NIST$_p$-3, SP-NIST$_p$-4, SP-NIST$_p$-5,
SP-NISTi$_p$-2, SP-NISTi$_p$-3, SP-NISTi$_p$-4, SP-NISTi$_p$-5 }

Table 7: Metrics based on Shallow Parsing

DP = { DP-O_c-$\star$, DP-O_c_a, DP-O_c_as, DP-O_c_aux, DP-O_c_be, DP-O_c_c, DP-O_c_comp, DP-O_c_det, DP-O_c_have, DP-O_c_n, DP-O_c_postdet, DP-O_c_ppspec DP-O_c_predet, DP-O_c_saidx, DP-O_c_sentadjunct, DP-O_c_subj, DP-O_c_that, DP-O_c_prep, DP-O_c_u, DP-O_c_v, DP-O_c_vbe, DP-O_c_xsaid, DP-HWC$_c$-1, DP-HWC$_c$-2, DP-HWC$_c$-3, DP-HWC$_c$-4, DP-HWC$_r$-1, DP-HWC$_r$-2, DP-HWC$_r$-3, DP-HWC$_r$-4, DP-HWC$_w$-1, DP-HWC$_w$-2, DP-HWC$_w$-3, DP-HWC$_w$-4, DP-HWCi$_c$-2, DP-HWCi$_c$-3, DP-HWCi$_c$-4, DP-HWCi$_r$-2, DP-HWCi$_r$-3, DP-HWCi$_r$-4, DP-HWCi$_w$-2, DP-HWCi$_w$-3, DP-HWCi$_w$-4, DP-O_l-$\star$, DP-O_l_1, DP-O_l_2, DP-O_l_3, DP-O_l_4, DP-O_l_5, DP-O_l_6, DP-O_l_7, DP-O_l_8, DP-O_l_9, DP-O_r-$\star$, DP-O_r_amod, DP-O_r_amount-value, DP-O_r_appo, DP-O_r_appo-mod, DP-O_r_as-arg, DP-O_r_as1, DP-O_r_as2, DP-O_r_aux, DP-O_r_be, DP-O_r_being, DP-O_r_by-subj, DP-O_r_c, DP-O_r_cn, DP-O_r_comp1, DP-O_r_conj, DP-O_r_desc, DP-O_r_dest, DP-O_r_det, DP-O_r_else, DP-O_r_fc, DP-O_r_gen, DP-O_r_guest, DP-O_r_have, DP-O_r_head, DP-O_r_i, DP-O_r_inv-aux, DP-O_r_inv-have, DP-O_r_lex-dep, DP-O_r_lex-mod, DP-O_r_mod, DP-O_r_mod-before, DP-O_r_neg, DP-O_r_nn, DP-O_r_num, DP-O_r_num-mod, DP-O_r_obj, DP-O_r_obj1, DP-O_r_obj2, DP-O_r_p, DP-O_r_p-spec, DP-O_r_pcomp-c, DP-O_r_pcomp-n, DP-O_r_person, DP-O_r_pnmod, DP-O_r_poss, DP-O_r_post, DP-O_r_pre, DP-O_r_pred, DP-O_r_punc, DP-O_r_rel, DP-O_r_s, DP-O_r_sc, DP-O_r_subcat, DP-O_r_subclass, DP-O_r_subj, DP-O_r_title, DP-O_r_vrel, DP-O_r_wha, DP-O_r_whn, DP-O_r_whp }

Table 8: Metrics based on Dependency Parsing

CP = { CP-O_c-$\star$, CP-O_c-ADJP, CP-O_c-ADVP, CP-O_c-CONJP, CP-O_c-FRAG, CP-O_c-INTJ, CP-O_c-LST, CP-O_c-NAC, CP-O_c-NP, CP-O_c-NX, CP-O_c-O, CP-O_c-PP, CP-O_c-PRN, CP-O_c-PRT, CP-O_c-QP, CP-O_c-RRC, CP-O_c-S, CP-O_c-SBAR, CP-O_c-SINV, CP-O_c-SQ, CP-O_c-UCP, CP-O_c-VP, CP-O_c-WHADJP, CP-O_c-WHADVP, CP-O_c-WHNP, CP-O_c-WHPP, CP-O_c-X, CP-O_p-#, CP-O_p-\$, CP-O_p-”, CP-O_p-(, CP-O_p-), CP-O_p-$\star$, CP-O_p-,, CP-O_p-., CP-O_p-:, CP-O_p-CC, CP-O_p-CD, CP-O_p-DT, CP-O_p-EX, CP-O_p-F, CP-O_p-FW, CP-O_p-IN, CP-O_p-J, CP-O_p-JJ, CP-O_p-JJR, CP-O_p-JJS, CP-O_p-LS, CP-O_p-MD, CP-O_p-N, CP-O_p-NN, CP-O_p-NNP, CP-O_p-NNPS, CP-O_p-NNS, CP-O_p-P, CP-O_p-PDT, CP-O_p-POS, CP-O_p-PRP, CP-O_p-PRP\$, CP-O_p-R, CP-O_p-RB, CP-O_p-RBR, CP-O_p-RBS, CP-O_p-RP, CP-O_p-SYM, CP-O_p-TO, CP-O_p-UH, CP-O_p-V, CP-O_p-VB, CP-O_p-VBD, CP-O_p-VBG, CP-O_p-VBN, CP-O_p-VBP, CP-O_p-VBZ, CP-O_p-W, CP-O_p-WDT, CP-O_p-WP, CP-O_p-WP\$, CP-O_p-WRB, CP-O_p-“, CP-STM-1, CP-STM-2, CP-STM-3, CP-STM-4, CP-STM-5, CP-STM-6, CP-STM-7, CP-STM-8, CP-STM-9, CP-STMi-2, CP-STMi-3, CP-STMi-4, CP-STMi-5, CP-STMi-6, CP-STMi-7, CP-STMi-8, CP-STMi-9 }

Table 9: Metrics based on Constituency Parsing

$$\text{NE} = \{ \quad \text{NE-}M_e\text{-}\star,\ \text{NE-}M_e\text{-ANGLE\_QUANTITY},\ \text{NE-}M_e\text{-DATE},$$
$$\text{NE-}M_e\text{-DISTANCE\_QUANTITY},\ \text{NE-}M_e\text{-LANGUAGE},$$
$$\text{NE-}M_e\text{-LOC},\ \text{NE-}M_e\text{-METHOD},\ \text{NE-}M_e\text{-MISC},$$
$$\text{NE-}M_e\text{-MONEY},\ \text{NE-}M_e\text{-NUM},\ \text{NE-}M_e\text{-ORG},\ \text{NE-}M_e\text{-PER},$$
$$\text{NE-}M_e\text{-PERCENT},\ \text{NE-}M_e\text{-PROJECT},\ \text{NE-}M_e\text{-SIZE\_QUANTITY},$$
$$\text{NE-}M_e\text{-SPEED\_QUANTITY},\ \text{NE-}M_e\text{-SYSTEM},$$
$$\text{NE-}M_e\text{-TEMPERATURE\_QUANTITY},\ \text{NE-}M_e\text{-WEIGHT\_QUANTITY},$$
$$\text{NE-}O_e\text{-}\star,\ \text{NE-}O_e\text{-}\star\star,\ \text{NE-}O_e\text{-ANGLE\_QUANTITY},$$
$$\text{NE-}O_e\text{-DATE},\ \text{NE-}O_e\text{-DISTANCE\_QUANTITY},$$
$$\text{NE-}O_e\text{-LANGUAGE},\ \text{NE-}O_e\text{-LOC},\ \text{NE-}O_e\text{-METHOD},$$
$$\text{NE-}O_e\text{-MISC},\ \text{NE-}O_e\text{-MONEY},\ \text{NE-}O_e\text{-NUM},$$
$$\text{NE-}O_e\text{-O},\ \text{NE-}O_e\text{-ORG},\ \text{NE-}O_e\text{-PER},$$
$$\text{NE-}O_e\text{-PERCENT},\ \text{NE-}O_e\text{-PROJECT},$$
$$\text{NE-}O_e\text{-SIZE\_QUANTITY},\ \text{NE-}O_e\text{-SPEED\_QUANTITY},$$
$$\text{NE-}O_e\text{-SYSTEM},\ \text{NE-}O_e\text{-TEMPERATURE\_QUANTITY},$$
$$\text{NE-}O_e\text{-WEIGHT\_QUANTITY} \quad \}$$

Table 10: Metrics based on <u>N</u>amed <u>E</u>ntities

SR = { SR-O_r, SR-O_{rv}, SR-N-$_v$, SR-O_v, SR-M_r-$\star$, SR-M_r-A0, SR-M_r-A1, SR-M_r-A2, SR-M_r-A3, SR-M_r-A4, SR-M_r-A5, SR-M_r-AA, SR-M_r-AM-ADV, SR-M_r-AM-CAU, SR-M_r-AM-DIR, SR-M_r-AM-DIS, SR-M_r-AM-EXT, SR-M_r-AM-LOC, SR-M_r-AM-MNR, SR-M_r-AM-MOD, SR-M_r-AM-NEG, SR-M_r-AM-PNC, SR-M_r-AM-PRD, SR-M_r-AM-REC, SR-M_r-AM-TMP, SR-M_{rv}-$\star$, SR-M_{rv}-A0, SR-M_{rv}-A1, SR-M_{rv}-A2, SR-M_{rv}-A3, SR-M_{rv}-A4, SR-M_{rv}-A5, SR-M_{rv}-AA, SR-M_{rv}-AM-ADV, SR-M_{rv}-AM-CAU, SR-M_{rv}-AM-DIR, SR-M_{rv}-AM-DIS, SR-M_{rv}-AM-EXT, SR-M_{rv}-AM-LOC, SR-M_{rv}-AM-MNR, SR-M_{rv}-AM-MOD, SR-M_{rv}-AM-NEG, SR-M_{rv}-AM-PNC, SR-M_{rv}-AM-PRD, SR-M_{rv}-AM-REC, SR-M_{rv}-AM-TMP, SR-O_r-$\star$, SR-O_r-A0, SR-O_r-A1, SR-O_r-A2, SR-O_r-A3, SR-O_r-A4, SR-O_r-A5, SR-O_r-AA, SR-O_r-AM-ADV, SR-O_r-AM-CAU, SR-O_r-AM-DIR, SR-O_r-AM-DIS, SR-O_r-AM-EXT, SR-O_r-AM-LOC, SR-O_r-AM-MNR, SR-O_r-AM-MOD, SR-O_r-AM-NEG, SR-O_r-AM-PNC, SR-O_r-AM-PRD, SR-O_r-AM-REC, SR-O_r-AM-TMP, SR-O_{rv}-$\star$, SR-O_{rv}-A0, SR-O_{rv}-A1, SR-O_{rv}-A2, SR-O_{rv}-A3, SR-O_{rv}-A4, SR-O_{rv}-A5, SR-O_{rv}-AA, SR-O_{rv}-AM-ADV, SR-O_{rv}-AM-CAU, SR-O_{rv}-AM-DIR, SR-O_{rv}-AM-DIS, SR-O_{rv}-AM-EXT, SR-O_{rv}-AM-LOC, SR-O_{rv}-AM-MNR, SR-O_{rv}-AM-MOD, SR-O_{rv}-AM-NEG, SR-O_{rv}-AM-PNC, SR-O_{rv}-AM-PRD, SR-O_{rv}-AM-REC, SR-O_{rv}-AM-TMP, SR-M_r-$\star$-b, SR-M_r-$\star$-i, SR-M_{rv}-$\star$-b, SR-M_{rv}-$\star$-i, SR-O_r-$\star$-b, SR-O_r-$\star$-i, SR-O_{rv}-$\star$-b, SR-O_{rv}-$\star$-i }

Table 11: Metrics based on Semantic Roles

DR = { DR-O_r-$\star$, DR-O_r-alfa, DR-O_r-card, DR-O_r-dr, DR-O_r-drs, DR-O_r-eq, DR-O_r-imp, DR-O_r-merge, DR-O_r-named, DR-O_r-not, DR-O_r-or, DR-O_r-pred, DR-O_r-prop, DR-O_r-rel, DR-O_r-smerge, DR-O_r-timex, DR-O_r-whq, DR-O_{rp}-$\star$, DR-O_{rp}-alfa, DR-O_{rp}-card, DR-O_{rp}-dr, DR-O_{rp}-drs, DR-O_{rp}-eq, DR-O_{rp}-imp, DR-O_{rp}-merge, DR-O_{rp}-named, DR-O_{rp}-not, DR-O_{rp}-or, DR-O_{rp}-pred, DR-O_{rp}-prop, DR-O_{rp}-rel, DR-O_{rp}-smerge, DR-O_{rp}-timex, DR-O_{rp}-whq, DR-STM-1, DR-STM-2, DR-STM-3, DR-STM-4, DR-STM-5, DR-STM-6, DR-STM-7, DR-STM-8, DR-STM-9, DR-STMi-2, DR-STMi-3, DR-STMi-4, DR-STMi-5, DR-STMi-6, DR-STMi-7, DR-STMi-8, DR-STMi-9, DR-O_r-$\star$-b, DR-O_r-$\star$-i, DR-O_{rp}-$\star$-b, DR-O_{rp}-$\star$-i, DR-STM-4-b, DR-STM-4-i }

Table 12: Metrics based on Discourse Representations

B Linguistic Processors and Tag Sets

B.1 Shallow Syntactic Parsing

Shallow parsing is performed using several state-of-the-art performance tools.

B.1.1 Part-of-speech Tagging

PoS and lemma annotation is automatically provided by the SVMTool (Giménez & Màrquez, 2004a; Giménez & Màrquez, 2004b)[35]. We use the Freeling (Carreras et al., 2004)[36] package only for lemmatization.

The SVMTool for English has been trained on the Wall Street Journal (WSJ) corpus (1,173K words). Sections 0-18 were used for training (912K words), 19-21 for validation (131K words), and 22-24 for test (129K words), respectively. 2.81% of the words in the test set are unknown to the training set. Best other results so far reported on this same test set are (Collins, 2002) (97.11%) and (Toutanova et al., 2003) (97.24%). Table 13 shows the SVMTool performance as compared to the TnT tagger. 'known' and 'unk.' refer to the subsets of known and unknown words, respectively. 'amb' refers to the set of ambiguous known words and 'all' to the overall accuracy.

	known	amb.	unk.	all.
TnT	96.76%	92.16%	85.86%	96.46%
SVMTool	97.39%	93.91%	89.01%	97.16%

Table 13: Performance of the SVMTool for English on the WSJ corpus

Table 14 and Table 15 show the PoS tag set for English, derived from the Penn Treebank[37] tag set (Marcus et al., 1993). Several coarse classes are included.

The SVMTool for Spanish has been trained on the 3LB [38] corpus (75K words). It was randomly divided into training set (59K words) and test set (16K words). 13.65% of the words in the test set are unknown to the training set. See results in Table 16.

Tag set for Spanish, derived from the PAROLE tag set, is shown in Table 17, Table 18 and Table 19.

B.1.2 Lemmatization

Word lemmas have been obtained by matching word-PoS pairs against two lemmaries available inside the Freeling package. The English lemmary contains lemmas for 185,201 different word-PoS pairs, whereas the Spanish lemmary contains lemmas for 1,039,365 word-PoS pairs.

B.1.3 Chunking

Partial parsing information (i.e., base phrase chunks) is obtained using the Phreco software based on global on-line learning via the Perceptron algorithm (Carreras et al., 2005).

English models have been trained on the Penn Treebank (300K words). We randomly split data into train (211,727 words), development (47,377 words) and test (40,039 words). Best performance

[35] `http://www.lsi.upc.es/~nlp/SVMTool/`
[36] `http://www.lsi.upc.es/~nlp/freeling/`
[37] `http://www.cis.upenn.edu/~treebank/`
[38] The 3LB project is funded by the Spanish Ministry of Science and Technology (FIT-15050-2002-244), visit the project website at `http://www.dlsi.ua.es/projectes/3lb/`.

Type	**Description**
CC	Coordinating conjunction, e.g., and,but,or...
CD	Cardinal Number
DT	Determiner
EX	Existential there
FW	Foreign Word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List Item Marker
MD	Modal, e.g., can, could, might, may...
NN	Noun, singular or mass
NNP	Proper Noun, singular
NNPS	Proper Noun, plural
NNS	Noun, plural
PDT	Predeterminer, e.g., all, both ... when they precede an article
POS	Possessive Ending, e.g., Nouns ending in 's
PRP	Personal Pronoun, e.g., I, me, you, he...
PRP$	Possessive Pronoun, e.g., my, your, mine, yours...
RB	Adverb. Most words that end in -ly as well as degree words like quite, too and very.
RBR	Adverb. comparative Adverbs with the comparative ending -er, with a strictly comparative meaning.
RBS	Adverb, superlative
RP	Particle
SYM	Symbol. Should be used for mathematical, scientific or technical symbols
TO	to
UH	Interjection, e.g., uh, well, yes, my...

Table 14: PoS tag set for English (1/2)

Type	**Description**
VB	Verb, base form subsumes imperatives, infinitives and subjunctives
VBD	Verb, past tense includes the conditional form of the verb to be
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner, e.g., which, and that when it is used as a relative pronoun
WP	Wh-pronoun, e.g., what, who, whom...
WP$	Possessive wh-pronoun
WRB	Wh-adverb, e.g., how, where why
#	
$	
”	
(	
)	Punctuation Tags
,	
.	
:	
“	

COARSE TAGS	
N	Nouns
V	Verbs
J	Adjectives
R	Adverbs
P	Pronouns
W	Wh- pronouns
F	Punctuation

Table 15: PoS tag set for English (2/2)

	known	amb.	unk.	all.
TnT	97.73%	93.70%	87.66%	96.50%
SVMTool	98.08%	95.04%	88.28%	96.89%

Table 16: Performance of the SVMTool for Spanish on the 3LB corpus

Type	**Description**
	NOUN
NC	Noun, Common
NP	Noun, Proper
	VERB
VAG	Verb, Auxiliary, Gerund
VAI	Verb, Auxiliary, Indicative
VAM	Verb, Auxiliary, Imperative
VAN	Verb, Auxiliary, Infinitive
VAP	Verb, Auxiliary, Participle
VAS	Verb, Auxiliary, Subjunctive
VMG	Verb, Main, Gerund
VMI	Verb, Main, Indicative
VMM	Verb, Main, Imperative
VMN	Verb, Main, Infinitive
VMP	Verb, Main, Participle
VMS	Verb, Main, Subjunctive
VSG	Verb, Semi-Auxiliary, Gerund
VSI	Verb, Semi-Auxiliary, Indicative
VSM	Verb, Semi-Auxiliary, Imperative
VSN	Verb, Semi-Auxiliary, Infinitive
VSP	Verb, Semi-Auxiliary, Participle
VSS	Verb, Semi-Auxiliary, Subjunctive
	ADJECTIVE
AO	Adjective, Ordinal
AQ	Adjective, Qualifier
AQP	Adjective, Qualifier and Past Participle
	ADVERB
RG	Adverb, General
RN	Adverb, Negative
	PRONOUN
P0	Pronoun, Clitic
PD	Pronoun, Demonstrative
PE	Pronoun, Exclamatory
PI	Pronoun, Indefinite
PN	Pronoun, Numeral
PP	Pronoun, Personal
PR	Pronoun, Relative
PT	Pronoun, Interrogative
PX	Pronoun, Possessive

Table 17: PoS tag set for Spanish and Catalan (1/3)

Type	**Description**
	ADPOSITON
SP	Adposition, Preposition
	CONJUNCTION
CC	Conjunction, Coordinate
CS	Conjunction, Subordinative
	DETERMINER
DA	Determiner, Article
DD	Determiner, Demonstrative
DE	Determiner, Exclamatory
DI	Determiner, Indefinite
DN	Determiner, Numeral
DP	Determiner, Possessive
DT	Determiner, Interrogative
	INTERJECTION
I	Interjection
	DATE TIMES
W	Date Times
	UNKNOWN
X	Unknown
	ABBREVIATION
Y	Abbreviation
	NUMBERS
Z	Figures
Zm	Currency
Zp	Percentage

Table 18: PoS tag set for Spanish and Catalan (2/3)

Type	Description
	PUNCTUATION
Faa	Fat Punctuation, !
Fc	Punctuation, ,
Fd	Punctuation, :
Fe	Punctuation, “
Fg	Punctuation, -
Fh	Punctuation, /
Fia	Punctuation,
Fit	Punctuation, ?
Fp	Punctuation, .
Fpa	Punctuation, (
Fpt	Punctuation,)
Fs	Punctuation, ...
Fx	Punctuation, ;
Fz	Punctuation, other than those

	COARSE TAGS
A	Adjectives
C	Conjunctions
D	Determiners
F	Punctuation
I	Interjections
N	Nouns
P	Pronouns
S	Adpositions
V	Verbs
VA	Auxiliary Verbs
VS	Semi-Auxiliary Verbs
VM	Main Verbs

Table 19: PoS tag set for Spanish and Catalan (3/3)

(F_1 = 93.72%) was obtained using averaged perceptrons up to epoch 8. Table 20 shows phrase chunking tag sets for English.

Type	**Description**
ADJP	Adjective phrase
ADVP	Adverb phrase
CONJP	Conjunction
INTJ	Interjection
LST	List marker
NP	Noun phrase
PP	Preposition
PRT	Particle
SBAR	Subordinated Clause
UCP	Unlike Coordinated phrase
VP	Verb phrase
O	Not-A-Phrase

Table 20: Base phrase chunking tag set for English

Models for Spanish have been trained on the 3LB corpus (95K words), randomly split into training (76,115 words) and test (18,792 words). Best performance (F_1 = 94.55%) was obtained using regular perceptrons after epoch 20. Table 21 shows phrase chunking tag sets for Spanish.

Type	**Description**
ADJP	Adjective phrase
ADVP	Adverb phrase
CONJP	Conjunction
INTJ	Interjection
NP	Noun phrase
PP	Preposition
SBAR	Subordinated Clause
VP	Verb phrase
AVP	Adjectival verb phrase
NEG	Negation
MORFV	Verbal morpheme
O	Not-A-Phrase

Table 21: Base phrase chunking tag set for Spanish and Catalan

B.2 Syntactic Parsing

Dependency parsing for English is performed using the MINIPAR[39] parser (Lin, 1998). A brief description of grammatical categories and relations may be found in Table 22 and Table 23.

[39] 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	Auxiliary verbs, e.g. should, will, does, ...
Be	Different 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 (the subject), 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 22: Grammatical categories provided by MINIPAR

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	relationship 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	relationship between a word and its adjunct 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
wha, whn, whp	wh-elements at C-spec positions (a|n|p)

Table 23: Grammatical relationships provided by MINIPAR

Constituency parsing for English is performed using the Charniak-Johnson's Max-Ent reranking parser (Charniak & Johnson, 2005)[40]. A description of the tag set employed is available in Table 24.

Type	**Description**
	Clause Level
S	Simple declarative clause
SBAR	Clause introduced by a (possibly empty) subordinating conjunction
SBARQ	Direct question introduced by a wh-word or a wh-phrase
SINV	Inverted declarative sentence, i.e. one in which the subject follows the tensed verb or modal
SQ	Inverted yes/no question, or main clause of a wh-question, following the wh-phrase in SBARQ
	Phrase Level
ADJP	Adjective Phrase
ADVP	Adverb Phrase
CONJP	Conjunction Phrase
FRAG	Fragment
INTJ	Interjection
LST	List marker
NAC	Not a Constituent; used to show the scope of certain prenominal modifiers within a NP
NP	Noun Phrase
NX	Used within certain complex NPs to mark the head of the NP
PP	Prepositional Phrase
PRN	Parenthetical
PRT	Particle. Category for words that should be tagged RP
QP	Quantifier Phrase (i.e. complex measure/amount phrase); used within NP
RRC	Reduced Relative Clause
UCP	Unlike Coordinated Phrase
VP	Verb Phrase
WHADJP	Wh-adjective Phrase
WHAVP	Wh-adverb Phrase
WHNP	Wh-noun Phrase
WHPP	Wh-prepositional Phrase
X	Unknown, uncertain, or unbracketable

Table 24: Clause/phrase level tag set for English

[40] `ftp://ftp.cs.brown.edu/pub/nlparser/`

B.3 Shallow Semantic Parsing

Named entities are automatically annotated using the BIOS Suite of Syntactico-Semantic Analyzers (Surdeanu et al., 2005)[41]. The list of NE types utilized is available in Table 25.

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 25: Named Entity types

Semantic role labeling is performed using the SwiRL Semantic Role Labeler (Surdeanu & Turmo, 2005; Màrquez et al., 2005)[42]. A list of SR types is available in Table 26.

B.4 Semantic Parsing

Semantic parsing is performed using the BOXER component (Bos, 2005) available inside the C&C Tools (Clark & Curran, 2004)[43]. BOXER elaborates DRS representations of input sentences parsed on the basis of a Combinatory Categorial Grammar (CCG) parser (Bos et al., 2004).

There are two types of DRS conditions:

basic conditions: one-place properties (predicates), two-place properties (relations), named entities, time expressions, cardinal expressions and equalities.

complex conditions: disjunction, implication, negation, question, and propositional attitude operations.

[41] `http://www.surdeanu.name/mihai/bios/`
[42] `http://www.surdeanu.name/mihai/swirl/`
[43] `http://svn.ask.it.usyd.edu.au/trac/candc`

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 26: Semantic Roles

Tables 27 to 31 describe some aspects of the DRS representations utilized. For instance, Tables 27 and 28 respectively show basic and complex DRS conditions. Table 29 shows DRS subtypes. Tables 30 and 31 show symbols for one-place and two-place relations.

Type	**Description**
pred	one-place properties (predicates)
rel	two-place properties (relations)
named	named entities
timex	time expressions
card	cardinal expressions
eq	equalities

Table 27: Discourse Representation Structures. Basic DRS-conditions

Type	**Description**
or	disjunction
imp	implication
not	negation
whq	question
prop	propositional attitude

Table 28: Discourse Representation Structures. Complex DRS-conditions

Type	Description
Types of anaphoric information	
pro	anaphoric pronoun
def	definite description
nam	proper name
ref	reflexive pronoun
dei	deictic pronoun
Part-of-speech type	
n	noun
v	verb
a	adjective/adverb
Named Entity types	
org	organization
per	person
ttl	title
quo	quoted
loc	location
fst	first name
sur	surname
url	URL
ema	email
nam	name (when type is unknown)
Cardinality type	
eq	equal
le	less or equal
ge	greater or equal

Table 29: Discourse Representation Structures. Subtypes

Type	Description
topic,a,n	elliptical noun phrases
thing,n,12	used in NP quantifiers: 'something', etc.)
person,n,1	used in first-person pronouns, 'who'-questions)
event,n,1	introduced by main verbs)
group,n,1	used for plural descriptions)
reason,n,2	used in 'why'-questions)
manner,n,2	used in 'how'-questions)
proposition,n,1	arguments of propositional complement verbs)
unit_of_time,n,1	used in 'when'-questions)
location,n,1	used in 'there' insertion, 'where'-questions)
quantity,n,1	used in 'how many')
amount,n,3	used in 'how much')
degree,n,1	
age,n,1	
neuter,a,0	used in third-person pronouns: it, its)
male,a,0	used in third-person pronouns: he, his, him)
female,a,0	used in third-person pronouns: she, her)
base,v,2	
bear,v,2	

Table 30: Discourse Representation. Symbols for one-place predicates used in basic DRS conditions

Type	Description
rel,0	general, underspecified type of relation
loc_rel,0	locative relation
role,0	underspecified role: agent,patient,theme
member,0	used for plural descriptions
agent,0	subject
theme,0	indirect object
patient,0	semantic object, subject of passive verbs

Table 31: Discourse Representation. Symbols for two-place relations used in basic DRS conditions