Generation Challenges 2011 Surface Realisation Shared Task: Documentation and Instructions for Participants

April 19, 2011

Contents

1 Overview

This is the documentation distributed to registered participants in the Surface Realisation Shared Task (SR Task, for short), held in Oct 2010–Sep 2011 as part of Generation Challenges 2011.

The following sections describe in detail the data, task and evaluation procedures of SR Task 2011, as well as how to participate and what to submit. A quick-start summary is provided in Section 7; it is intended for quick reference only, and does not provide complete information about the SR Task 2011.

2 Data

2.1 Two levels of input representation: Shallow and Deep

The SR Task data consists of two types of input representations—one shallow, one deep. In both, sentences are represented as sets of unordered labeled dependencies (with the exception of named entities (see Section 2.2.4 below), which are ordered). The shallow input representation is a more 'surfacey', syntactic represention of the sentence. The deep(er) input type is closer to a semantic, more abstract, representation of the meaning of the sentence.

The data has been constructed by post-processing the CoNLL 2008 Shared Task data (Surdeanu et al., 2008). For the preparation of the CoNLL-08 Shared task data, the Penn WSJ Treebank was converted to syntactic dependencies via the LTH Constituent-to-Dependency Conversion Tool for Penn-style Treebanks (Pennconverter) (Johansson and Nugues, 2007). The resulting dependency bank was then merged with the Nombank (Meyers et al., 2004) and Propbank corpora (Palmer et al., 2005). Named entity information from the BBC Entity Type corpus was also integrated into the CoNLL-08 data. Our shallow level representation is based on the Pennconverter dependencies. The semantic level representation is derived from the merged Nombank, Propbank and syntactic dependencies in a process similar to the graph completion algorithm outlined in (Bohnet et al., 2010) (see Section 2.2, in particular Subsection 2.2.2, for differences).

One motivation behind having two levels of representation is to be inclusive. Existing sentence realisation systems have varying levels of surface detail in their expected input. With two levels in the shared task, it is more likely that people will find an input that suits their system. In addition, systems that require a surface realisation component (e.g. question answering, text summarisation, transfer-based machine translation) can choose from two possible levels of realiser input, thus making the shared task potentially useful to a wider audience.

Ideally, the semantic representation should abstract away from the surface features of the sentence. It should be less specified than the syntactic representation with a one-to-many relationship between the deep and surface level representation, reflecting the variety of ways an underlying meaning can be realised (though the syntactic level too can have many different realisations). In this year's pilot task, due to its method of construction, the semantic data contains many 'surfacey' details, including some function words, punctuation markers and syntactic edges, which in future years we are planning to remove.

2.2 Representation Details

In this section, we give an overview of the shallow and deep representations, with Section 2.2.6 and Table 2 detailing the representation format.

2.2.1 Shallow

The shallow data consists of unordered syntactic dependency trees. Each word and punctuation marker from the original sentence is represented as a node in a syntactic dependency tree.

Nodes

The node information consists of a word's lemma, a coarse-grained POS-tag, and, where appropriate, number, tense and participle features and a sense tag id (as a suffix to the lemma). In addition, two punctuation features encode the quotation and bracketing information for the sentence.

The POS-tag set is slightly less fine-grained than the Penn POS-tag set. We have removed the distinction between VBP and VBZ for example, so that determining agreement is a task left to the realiser.

Edges

Edges between nodes are labeled with the syntactic labels produced by the Penncoverter. See Table 1 for a summary description of the label set (taken from Surdeanu et al. (2008)). In addition to the *atomic* labels in Table 1, edges can be labeled with *non-atomic* labels, which consist of multiple atomic labels (see Surdeanu et al. (2008) for details).

Label	Description
ADV	General Adverbial
AMOD	Modifier of adjective or adverbial
APPO	Apposition
BNF	Benefactor complement *(for)* in dative shift
CONJ	Second conjunct (dependent on conjunction)
COORD	Coordination
DEP	Unclassified
DIR	Adverbial of direction
DTV	Dative compelent *(to)* in dative shift
EXT	Adverbial of extent
EXTR	Extraposed element in cleft
HMOD	Token inside a hyphenated word (dependent on the head of the hyphenated word).
HYPH	Token part of a hyphenated word (dependent on the preceeding part of the hyphenated word)
IM	Infinitive verb (dependent on infinitive marker *to*
LGS	Logical subject of a passive verb
LOC	Locative adverbial or nominal modifier
MNR	Adverbial of manner
NAME	Name-internal link
NMOD	Modifier of nominal
OBJ	Object
OPRD	Predicative complement of raising/control verb
P	Punctuation
PMOD	Modifier of preposition
POSTHON	Posthonorific modifier of nominal
PRD	Predictive complement
PRN	Parenthetical
PRP	Adverbial of purpose or reason
PRT	Particle (dependent on verb)
PUT	Complement of the verb *put*
SBJ	Subject
SUB	Subordinated clause (dependent on subordinating conjunction)
SUFFIX	Possessive suffix (dependent on possessor)
SROOT	Root
TITLE	Title (dependent on name)
TMP	Temporal adverbial or nominal modifier
VC	Verb Chain
VOC	Vocative

Table 1: Atomic syntactic labels.

Note on Long Distance Dependencies

The Pennconverter handles some long distance dependencies, such as *wh*-movement and topicalization. However, as the output is strictly trees, not graphs, there are no multi-headed dependencies. Take, for example, the right node raising construction: 'He commissions and splendidly interprets fearsome contemporary scores'. Ideally, *scores* should have two heads: *commissions* and *interprets*. Figure 1 illustrates the Pennconverter output for the sentence. A post-processing step which adds some of these missing relations, producing graph representations with multiple headed dependencies, is something we would like to investigate for future work.

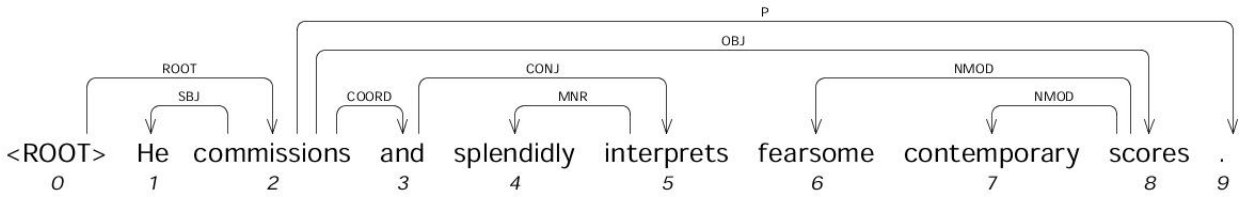

Figure 1: Example of right node raising, Pennconverter output (extract from wsj_0207)

2.2.2 Deep

The deep representation is in the form of dependency graphs and is not restricted to tree structures.

Nodes

Information at each node consists of a word's lemma, and where appropriate, number, tense and participle features and a sense tag id (as a suffix to the lemma). Two punctuation features encode the quotation and bracketing information for the sentence. Unlike in the shallow representation, there is no POS-tag information.

In a step toward removing punctuation from the deep representation, we have removed commas from the deep representation.[1] In addition, some function words (specifically, that-complementizers and TO infinitives) have been removed. For the future, we intend to remove further function words, such as, for example, relative pronouns and, possibly, case-marking prepositions.

Edges

Semantic edges are labeled with semantic labels taken from the Propbank and Nombank semantic roles. These are of the form $A0$, $A1$ .., An.[2]

Where the PropBank/NomBank relations results in an unconnected structure, the graph is connected with edges from the corresponding syntactic tree, with the syntactic labels produced by the Penncoverter.

Some of these Pennconverter labels have been modified slightly in an effort to make them more general. See Table 2 for details. In the case of NMOD and AMOD, the syntactic head is typically a semantic argument of its modifier; accordingly, these syntactic relations are replaced with an AINV (Argument INVerse) semantic relation. The direction of Pennconverter edges adopted in the deep representation remains unchanged.

2.2.3 Tokenisation

The tokenisation of the SR Task data follows that of the CoNLL data, which differs from the Penn Treebank tokenisation. Hyphenated words are split and dependencies between the split tokens are given. For example, *prime-time* is represented as three tokens with the dependencies: $[time]_{HMOD} \rightarrow [prime]_{HYPH} \rightarrow [-]$.

[1] Though the vast majority of the 47,921 commas in the original CoNLL training set were removed from the deep representation, there remain 55 occurrences of commas. These were cases where the comma had dependent nodes. It is our intention to remove these from future versions of the data.

[2] Propbank defines semantic roles on a verb by verb basis because of the difficulty of defining a universal set of labels that would cover all types of predicates. For a particular verb A0 is generally an argument exhibiting features of an Agent while A1 is Patient or Theme. See Palmer et al. (2005) for details.

2.2.4 Named Entities

Named entity annotations from the BBN Entity Type corpus were used to derive NAME dependencies in the CoNLL corpus. For the SR Task data we have numbered all NAME dependencies with the order they appear in the original sentence because, arguably, the ordering of words in named entities is not a task that should be left to a surface realizer.

2.2.5 Coordination

Following the CoNLL format, the first conjunct is the head of coordinate structures in both shallow and deep representations. All other conjuncts, and the coordinating conjunction, are descendants of the leftmost conjunct. The order of the conjuncts is encoded in the dependency structure, e.g. $[A]_{COORD} \rightarrow [B]_{COORD} \rightarrow [and]_{CONJ} \rightarrow [C]$ for the list *A, B and C*. As with named entities, in most cases the surface realiser should not be expected to make the right choice of conjunct order (e.g. 'The top three finishers were A, B and C, respectively'). The representation of coordination will be revisited in future years.

2.2.6 Data Format

The data format for the semantic graphs and shallow dependency trees follows the general rules:

- Before each graph is a line with the graph number (e.g. sentId=11055).
- Each line in the graph represents a single node and consists of at least three fields and a maximum of 10 fields.
- After each graph representation we include a line containing the original sentence, followed by a blank line (the test sets will not include the sentence).

The graph format is as follows:

RELATION ID PARENT ID LEMMA[.sensetagID] [CPOS=POStag] [num=sg|pl] [tense=past|pres] [partic=past|pres] [quoted=d*s*] [bracket=r*c*]

Each line contains at least the four fields RELATION, ID, PARENT ID and LEMMA[.sensetagID], except in the case of node X with multiple heads. In such cases a line for node X is printed for each $head \rightarrow nodeX$ relation. The first time this occurs the full information for node X is printed. For subsequent occurrences only the relation label, the node ID, and the parent node id are printed. Note that, as the syntactic representations are strictly trees, multiple heads will only occur in the deep representation. Table 2 describes the fields for shallow and deep representations in some detail.

For maximum human and machine readability, the dependency structure of the graphs is reflected both through tabular indentation and the ID and PARENT ID fields.

2.3 Differences in format between this release and the sample data

For the sake of completeness, we briefly summarise the changes we have made to the data format since the distribution of the sample data in March 2011.

One difference between the sample data and the latest release described in this document (which includes the updated sample data), is that the original sentence is now included after each graph (whereas before the original sentences were in a separate file). Futhermore, there are now two extra fields (QUOTED and BRACKET) for encoding information on quotations and brackets. For details of the latter please refer to Table 2.

2.4 Conversion script for converting CoNLL'08 format into SR Task format

We have decided not to distribute the conversion script for converting CoNLL'08-formatted data into SR-Task-formatted data during the competition. However, if you have extra data that you would like to convert to SR Task format (and which is not part of the CoNLL'08 data), we are happy to convert it for you. Please contact us on nlg-stec@itri.brighton.ac.uk and allow two working days for the converted data to be returned to you.

Name	Description/Comments
RELATION (shallow)	Syntactic dependency relations. NAME dependencies are numbered with order information. The root of the tree has relation SROOT.
RELATION (deep)	Semantic relations when available. Otherwise, they are the shallow relations, some of which have been simplified as follows: $NMOD\|AMOD \rightarrow AINV$ $HMOD \rightarrow MOD$ $PMOD \rightarrow A1$ Sentences have a single root, marked with relation SROOT.
ID	Token id of the node, starts at 1 for each new sentence
PARENTID	Token id of the parent of this node
LEMMA[.sensetagID]	Lemma with, when available, a sense tag id suffix. The lemma and sense tag id are the lemma and roleset id extracted from propbank/nombank. When this information is unavailable the lemma is the predicted lemma extracted from the CoNLL-08 data set.
CPOS (shallow)	Hand-annotated coarse grained POS tag (from PTB) $VBD\|VBN\|VBP\|VBZ \rightarrow VB$ $NNS \rightarrow NN$ $NNPS \rightarrow NNP$ all other POS tags $\rightarrow$ original hand-annotated PTB POS tag
NUM	Feature for nouns only. Values are singular or plural - derived from hand-annotated PTB POS tags. $NN\|NNP \rightarrow singular$ $NNS\|NNPS \rightarrow plural$
TENSE	Feature for verbs only. Values are past or pres(ent) - derived from hand-annotated PTB POS tags. $VBD \rightarrow past$ $VBP\|VBZ \rightarrow present$
PARTIC	Feature for participle tense derived from hand-annotated PTB POS tags (note: partic=pres could indicate a present participle or gerund). $VBN \rightarrow past$ $VBG \rightarrow pres$
QUOTED	Feature for indicating whether the node is quoted in the original sentence. $d = doublequoted$, $s = singlequoted$. This feature value can consist of any number of d's followed by any number of s's. Multiple d's or s's occur when the node is embedded inside more than one quotation mark. Take for example the sentence: He added :" *Every paper company management has to be saying to itself , ' Before someone comes after me , I 'm going to go after somebody . ' "* The node corresponding to *paper* will have feature $quoted = d$ and the node for word *someone* will have $quoted = ds$.
BRACKET	Feature for indicating whether the node is inside brackets in the original sentence. $r = round\ brackets$, $c = curly\ brackets$. In a similar fashion to the QUOTED feature, this feature value can consist of any number of r's followed by any number of c's.

Table 2: Field descriptions for Shallow and Deep Representations.

2.5 TODO

It is our intention to modify the input representations for use in future editions of the SR Task. What follows is our current TODO list. We encourage feedback and suggestions of further ways to improve on the representation.

- Remove more function words (e.g. relative pronouns) from the deep representation.
- Investigate the possibility of changing the representation of coordination, making the conjunct head.
- Refine, and possibly remove, some of the syntactic labels that occur in the semantic graphs.
- Create a capitalisation feature to encode case information for proper nouns and named entities on the basis that in the data-to-text case, one is likely to be inserting names like 'New York' and 'DPC Acquisition Partners' into the input representations for a realizer, with their desired capitalisation.
- Remove remaining commas and other punctuation markers from deep representation.

3 Task Definition

The task is to generate one or more surface strings for each dependency structure in the set of shallow and/or deep input representations. As described in Section 5, outputs will be evaluated in both single-best and n-best scenarios.

As described in the following sections, submitted test data outputs must identify exactly one surface string as the single best system output.

In addition, submitted test data outputs may identify up to 4 more strings, in ranked order, as further system outputs.

All participating systems will be evaluated in single-best mode. Teams can additionally elect to have their systems evaluated in 5-best mode. Note that in this mode, if for any given test data item, a system produces less than 5 outputs, all missing outputs will be scored 0 (1 for TER). Similarly, if there are duplicates, each duplicate (but not of course the first occurrence) will also be scored 0 (1 for TER).

The training and development data may be used to develop, train and tune realisation models. The test data will have the same format as the training and development data, except that the original surface strings will not be present.

Additional resources may be used in the realisation task, but reasonable care must be taken to ensure that they do not include any information from the test data. For example, n-gram models may be trained on the text in the training data as well as from other resources, but these other resources should not overlap with Sections 23 and 24 of the Penn Treebank.

For this year's task, the original tokenisation should be retained, e.g. punctuation should remain as separate tokens. For evaluation purposes we will ignore all surface string capitalisation.

Participants' reports should state clearly what data was used in training and tuning systems, how it was obtained and converted. This holds for both the data supplied by the SR Task organisers and additional data obtained from elsewhere.

4 Test Data Output Format

System output should be put in a single XML file whose format is illustrated in Figure 2. The file format is similar to ones used in MT evaluation, modified to include n-best results. The `setid` attribute should be either `genchal2011.sr.shallow` or `genchal2011.sr.deep`. The `sysid` should be the team ID you registered with, e.g. `BU` (or `BU-1`, `BU-2` ...in the case of multiple teams from the same site). Multiple submissions from each team may be submitted for automatic scoring, in which case the `sysid` for the primary submission should be suffixed with `.1`, and other submissions with `.2`, `.3`, etc.

The output for each sentence should be enclosed in segment tags `<seg id="1">...</seg>` where the id is the same as the `sentId` in the input file. The single-best output should come first and be enclosed in `<best>...</best>` tags. After the single-best output, up to four rank-ordered additional outputs may be given within `<next>...</next>` tags.

As shown in the output for sentence 3 in the figure, output should retain tokenisation as given in the input, including punctuation as separate tokens and hyphenated words split into separate tokens. For the purpose of evaluation, lower-case/upper-case distinctions will be ignored.

```
<tstset trglang="en" setid="genchal2011.sr.deep" sysid="BU-1.1">
<seg id="1">
<best>the economy ’s temperature will be taken from several vantage
points this week , with readings on trade , output , housing and
inflation .</best>
<next>the temperature of the economy will be taken from several vantage
points this week , with readings on trade , output , housing and
inflation .</next>
<next>this week the economy ’s temperature will be taken from several
vantage points , with readings on trade , output , housing and
inflation .</next>
<next>this week the temperature of the economy will be taken from several
vantage points , with readings on trade , output , housing and
inflation .</next>
<next>this week , the economy ’s temperature will be taken from several
vantage points , with readings on trade , output , housing and
inflation .</next>
</seg>
...
<seg id="3">
<best>the trade gap is expected to widen to about $ 9 billion from
july ’s $ 7.6 billion , according to a survey by mms international ,
a unit of mcgraw - hill inc. , new york .</best>
<next>...</next>
...
</seg>
...
</tstset>
```

Figure 2: File format for test data

5 Evaluation Procedures

5.1 Training, Development and Test Sets

We are following the main data set divisions of the CoNLL'08 data. However, we have removed 300 randomly selected sentences in chunks of 5 consecutive sentences for use in human evaluations. Of these, we will use a total of 100 sentences this year and will hold back the remaining sentences for use in future editions of the SR Shared Task.

1. Sample data: selected from PTB Section 24; note that the development set includes the sample data.
2. Training set: PTB Sections 02–21.
3. Development set: 1,034 sentences from PTB Section 24 (less 300 sentences for use in human evaluations in this and future editions of the SR Shared Task; see also below).
4. Test set for automatic evaluation: PTB Section 23.
5. Test set for human evaluation: 100 sentences in chunks of 5 consecutive sentences, randomly selected (and removed) from PTB Section 24.

Note that a small number of sentences from the original WSJ sections were not included in the CoNLL-08 data (and are thus not included in the SR Task data) due to difficulties in merging the various data sets (for example Section 23 contains 2,399 sentences instead of the original set of 2,416 sentences).

5.2 Evaluation Criteria

The measures identified as important for evaluation of surface realization output in previous work include:

C1 Adequacy (preservation of meaning).

C2 Fluency (grammatical and idiomatic English).

C3 Clarity/readability, some aspects of which can typically only be measured in context.

C4 Similarity to human-produced realisations, aka Humanlikeness.

C5 Task effectiveness.

This particular shared task has no easily identifiable aspect of task effectiveness, so we will focus on measuring the other four criteria.

5.3 Automatic Evaluation Metrics (C4)

We will compute scores to assess C4 using the following metrics:

1. BLEU-4 (implementation from http://www.itl.nist.gov/iad/mig//tests/mt/2009/): n-gram similarity.
2. NIST-4 (implementation from http://www.itl.nist.gov/iad/mig//tests/mt/2009/): n-gram similarity weighted in favour of less frequent n-grams which are taken to be more informative.
3. METEOR (implementation from http://www.cs.cmu.edu/ alavie/METEOR/): lexical similarity based on exact, stem, synonym, and paraphrase matches between words and phrases.
4. TER (implementation from http://www.umiacs.umd.edu/ snover/tercom/): messures the number of edits required to change a system output into one of the references (in contrast to the above three metrics, the best TER score is 0, and the higher the score, the worse it is).

Text normalisation: We will apply some simple text normalization (e.g. lowercasing) to system outputs before scoring them with the automatic metrics. This will be described in detail in the results report.

Use of automatic metrics in n-best, ranked system outputs scenario: We will compute a single score for all outputs by computing their weighted sum, where a weight w_i is assigned to the ith system output in inverse proportion to its rank r_i:

$$w_i = \frac{K - r_i + 1}{\sum_{j=1}^{K} K - r_j + 1}$$

For a small subset of the test data we will obtain additional alternative realisations via Mechanical Turk (as done e.g. by Bloodgood and Callison-Burch (2010)). Based on the experiences in that paper and in some other recent work, it seems likely that good alternative realizations can be obtained using Mechanical Turk. For example, in the AT&T Analytics project, turkers transcribe noisy speech; in the Rochester AudioWiz and VizWiz projects,[3] turkers identify objects in images and transcribe audio input. We intend to release the multiple paraphrases of the test sentences we obtain in this way as a valuable resource to the community after the evaluation is over.

5.4 Human Evaluation Experiments (C1, C2, C3)

Our intention is to carry out the following two types of human-assessed evaluation experiments. If carrying out both proves infeasible, we will carry out just the first one.

1. Preference Judgement Experiment (C2, C3): Collect preference judgments using an existing evaluation interface (Belz and Kow (2010)) and directly recruited evaluators. We will present sentences in the context of a chunk of 5 consecutive sentences (see above) to the evaluators, and ask for separate judgments for Clarity and Fluency.

2. HTER (Snover et al. (2006), Sripada et al. (2005)): Human evaluators are asked to post-edit the output of a system, and the edits are then categorized and counted. Crucial to this evaluation method is the construction of clear instructions for evaluators. Human effort is also required to categorize and count the edits; in the SR Task case, we will categorize edits as relating to Adequacy, Fluency and/or Clarity; we will also consider further subcategorizations.

Note that due to limited time and resources we are planning to carry out human evaluations for just the single-best scenario.

5.5 Missing Outputs

5.5.1 In single-best evaluation mode

All participating systems will be evaluated in single-best mode. It is likely that some systems will not produce any outputs for some test data inputs, and a clear policy is needed of what happens in the single-best evaluation when a system fails to produce any output for certain inputs. As a general principle, systems are required to submit at least one output for each test item. We will report single-best scores in two ways: one just using the scores on the items where the system has produced an output (and reporting coverage), and one using a score of 0 for such no-output items (1 for TER).

5.5.2 In n-best evaluation mode

Teams can additionally elect to have their systems evaluated in 5-best mode. In this mode, if for any given test data item, a system produces less than 5 outputs, all missing outputs will be scored 0 (1 for TER). Similarly, if there are duplicates, each duplicate (but not of course the first occurrence) will also be scored 0 (1 for TER).

5.6 Development Set Scores Computed by Participants

We will provide evaluation scripts to participants so they can perform automatic evaluations on the development data. These scores serve two purposes. Firstly, development data scores must be included in participants' reports. Secondly, participants may wish to use the evaluation scripts in developing and tuning their systems.

See also What To Submit Section below.

[3] http://hci.cs.rochester.edu/viswiz

5.7 Reporting of Scores and Analysis of Results by Organisers

We will report per-system results separately for the automatic metrics (4 sets of results), and for the human metrics (2 sets of results). For each set of results, we will report single-best and n-best results. For single-best results, we will furthermore report results both with and without missing outputs (see previous section). We will rank systems, and report significance of per-system differences using bootstrap resampling where necessary (Zhang and Vogel (2010), Koehn (2004)). We will separately report correlation between human and automatic metrics, and between automatic metrics.

5.8 Restriction on Number of Systems per Team for Human Evaluations

Participants who enter multiple systems in the SR Task may be required to select a subset of their systems for inclusion in the human evaluation for this task. The exact number will depend on how many systems will be submitted by participants in total.

Note that we will carry out automatic evaluations for any number of submitted systems.

5.9 Evaluation Metrics Track

We are also running a separate Evaluation Metrics Track (with its own documentation and data sets). In this track, any creator of an evaluation metric can use their metric on the shared task data, which will be made available for the metric developers after submission by the system developers (and on an anonymized basis—System 1, System 2 etc.). This will be run on a competitive basis, in a similar fashion to the main SR Task track. Participants in the evaluation metrics track must submit their outputs (sets of metric scores) and reports by the given deadline (see separate documentation). We will evaluate the outputs in terms of their correlation with the human scores (separate rank tables for each), and present the results in a separate report.

5.10 TODO

For future years of the SR Task, we will probably want to evaluate: (1) Other genres of text, i.e. non-newspaper; (2) Other types of discourse, e.g. surface realization for animated agents; and (3) Other languages.

We are also looking into the possibility of having new text annotated in the PTB style, to enable us to have genuinely unseen test data.

6 Participation and Submission

6.1 What/how to submit

Teams may participate in either one or both of the SR Task 2011 subtasks (SR-Shallow and SR-Deep). The following needs to be submitted by each participating team:

1. A single report describing the team's method(s) for SR-Deep and/or SR-Shallow, and reporting scores computed on the development set using the software provided by us; and

2. Outputs for the SR Task 2011 Test Set produced with the method(s) described in the report, and conforming to the output format described in Section 4.

Reports can be submitted at any time during the submission period (see Section 6.4), by uploading the report file on the submission webpage (accessible through the SR Task 2011 homepage). Once a report has been received, the relevant test data inputs become available for download. From the time of download, the team has 48 hours to submit the test data outputs.

6.2 Guidelines for reports

Each team needs to submit a single report, describing the method(s) they are submitting outputs for. The title of the report should contain the ID of the team and the ID of the system(s) being described (if the team ID is contained in the system ID, then just including the system ID is enough). The report should include scores computed on the development set.

Reports should state clearly what data was used in training and tuning systems, how it was obtained and converted. This holds for both the data supplied by the SR Task organisers and additional data obtained from elsewhere.

Reports should be submitted in Adobe PDF format, and should follow the ENLG'11 guidelines (see http://talc.loria.fr/Call-for-Papers.html). Reports must not exceed 2 (two) pages in length. A third page may contain bibliographic references only. *Please do not decrease the font size from the default provided.*

The results report by the organisers will contain a detailed description of the SR Task and data sets, so there is no need to describe these in detail in the participants' reports.

Note that the reports submitted should be final, camera-ready versions. Changes to the reports can subsequently only be made in exceptional circumstances.

NB: A team submitting to both SR-Deep and SR-Shallow, or multiple methods for one or both of these, should prepare a single report describing all methods.

6.3 Proceedings and presentations

Participants' reports will be included in the proceedings of the ENLG'11 Conference. Reports will not undergo a selection procedure with multiple reviews, but the organisers reserve the right to reject material which is not appropriate given the participation guidelines.

Participants are strongly encouraged, but not required, to attend the ENLG'11 Conference. Those participants who are able to attend may be invited to contribute to the presentation of results at the Generation Challenges 2011 Special Session at ENLG'11.

NB: All participants are required to prepare a poster based on their paper for inclusion in the Generation Challenges 2011 Poster Session.

6.4 Dates

18 Oct 2010	Announcement and Call for Expressions of Interest.
11 Mar 2011	Call for Pre-Registration and release of sample data.
31 Mar 2011	General Call for Participation and release of complete SR Task data.
01 Jul–01 Aug 2011	Test data submission period; 4-step submission process: 1. Fill in submission form (available on website from 01 Jul). 2. Upload 2-page paper (with a possible third page of references only) describing approach and reporting development set results. 3. Download test data (inputs only). 4. Submit test data outputs at the latest 48 hours after download, but in any case no later than 01 Aug.
01 Aug 2011	Final deadline for submission of test data outputs
01 Aug–01 Sep 2011	SR Task 2011 Evaluation period
28, 29 or 30 Sep 2011	Generation Challenges meeting at ENLG'11

7 Quick-start Summary

This section provides a very brief summary of guidelines and requirements for the SR Task 2011. This is not intended as a substitute for the detailed descriptions of guidelines and requirements in the preceding sections of this document which are the only descriptions that will be taken as definitive.

Participants should use the training/development data distributed for SR Task 2011 to create their systems.

The test data outputs to be submitted to SR Task 2011 should be in the format described in Section 4.

To submit: (i) 2-page report describing the method(s), and reporting scores computed on the Development Set with the software provided by the organisers; (ii) outputs for the test set.

Test data will be evaluated by the organisers using BLEU, NIST, METEOR and TER, in addition to direct human assessment of Referential Clarity and Fluency by preference judgments, and of Adequacy, Fluency and Clarity by a procedure similar to HTER.

8 Organisers and Contacts

Organising Team:

Anja Belz, NLTG, University of Brighton, UK
Josef van Genabith, CNGL, Dublin City University, Ireland
Deirdre Hogan, CNGL, Cublin City University, Ireland
Amanda Stent, AT&T Labs Research Inc., US
Mike White, Department of Linguistics, The Ohio State University, US

Additional members of Common-ground Input Representation Working Group:

Bernd Bohnet, IMS, University of Stuttgart, Germany
Johan Bos, Groningen University, Netherlands
Aoife Cahill, IMS, University of Stuttgart, Germany
Charles Callaway, University of Haifa, Israel
Pablo Gervás, Universidad Complutense de Madrid, Spain
Stephan Oepen, University of Oslo, Norway
Leo Wanner, Information and Communication Technologies, UPF, Barcelona, Spain

Contacts:

SR Task contact email: nlg-stec@itri.brighton.ac.uk
SR Task website: http://www.nltg.brighton.ac.uk/research/sr-task

References

Belz, A. and E. Kow (2010). Comparing Rating Scales and Preference Judgements in Language Evaluation. In Proceedings of the 15th International Natural Language Generation Conference (INLG'10). pp. 7–15.

Bloodgood, M. and C. Callison-Burch (2010). Using Mechanical Turk to build machine translation evaluation sets. In Proc. NAACL-HLT'10.

Bohnet, Bernd, Leo Wanner, Simon Mille and Alicia Burga (2010). Broad Coverage Multilingual Deep Sentence Generation with a Stochastic Multi-Level Realizer. In Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, China.

Johansson, Richard and Pierre Nugues (2007). Extended Constituent-to-Dependency Conversion for English. In Joakim Nivre, Heiki-Jaan Kaalep, Kadri Muischnek and Mare Koit (eds.), Proceedings of NODALIDA 2007. Tartu, Estonia, pp. 105–112.

Koehn, Philipp (2004). Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin and Dekai Wu (eds.), Proceedings of EMNLP 2004. Barcelona, Spain: Association for Computational Linguistics, pp. 388–395.

Meyers, Adam, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young and Ralph Grishman (2004). The Nombank Project: An Interim Report. In NAACL/HLT Workshop Frontiers in Corpus Annotation.

Palmer, Martha, Daniel Gildea and Paul Kingsbury (2005). The Proposition Bank: A Corpus Annotated with Semantic Roles. In Computational Linguistics Journal. pp. 71–105.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla and John Makhoul (2006). A study of translation edit rate with targeted human annotation. In In Proceedings of Association for Machine Translation in the Americas. pp. 223–231.

Sripada, Somayajulu, Ehud Reiter and Lezan Hawizy (2005). Evaluation of an NLG System using Post-Edit data: Lessons Learnt. In Proceedings of European Natural Language Generation Workshop (ENLG'05). pp. 133–139.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez and Joakim Nivre (2008). The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In Proceedings of the Twelfth Conference on Computational Natural Language Learning. Manchester, UK.

Zhang, Ying and Stephan Vogel (2010). Significance tests of automatic machine translation evaluation metrics. Machine Translation 24, 51–65.