SENSEVAL-3: Third International Workshop on the Evaluation of Systems
for the Semantic Analysis of Text, Barcelona, Spain, July 2004
Association for Computational Linguistics

A First Evaluation of Logic Form Identification Systems

Vasile Rus
Department of Computer Science
Indiana University
South Bend, IN 46634
vasile@cs.iusb.edu

Abstract

This paper presents a first experience with evaluating sytems that address the issue of Logic Form Identification (LFi). A Gold Standard approach was used in which experts provide solutions to test data. The expert solutions, the gold standard, are then compared against outputs from participanting systems and different metrics observed. We proposed a few novel metrics, including precision and recall, that are further used to provide comparative results. The test data included 4155 arguments and 2398 predicates grouped in 300 sentences.

1 Introduction

The goal of a Logic Form Identification (LFi) task is to evaluate the performance of different methods addressing the issue of LFi. The Logic Form (LF) that we use is a flat, scope-free first order logic representation that embeds lexical and syntactic information. Given a set of English sentences, participating systems were supposed to return the sentences in Logic Form as in the example below.

Input: *The Earth provides the food we eat every day.*
Output: `Earth:n_(x1) provide:v_(e1, x1, x2) food:n_(x2) we(x3) eat:v_(e2, x3, x2; x4) day:n_(x4)`

The general approach adopted for evaluation was a *gold standard* approach in which the test data is first correctly mapped onto its corresponding LF by a team of experts and then this correct LF is automatically compared against outputs provided by participating sytems.

2 General Guidelines

The Logic Form of a sentence is the conjunction of individual predicates, where the relationships among them are expressed via shared arguments. Predicates are generated for all content words such as nouns, verb, adjectives and adverbs. Pronouns are treated as independent nouns. Prepositions and conjunctions are also mapped onto predicates that capture the relation between the prepositional object and the constituent to which it is attached and the relation among the coordinated entities, respectively.

There are two types of arguments: *e* - for events, *x* - for entities. For the sentence presented above we have two events - *e1, e2* corresponding to each verb/action in the sentence and four entities - *x1, x2, x3, x4* corresponding to the heads of the base noun phrases (NP). Each verb predicate has the second argument set to the corresponding logical subject and the third argument to its direct object. The remaining slots for a verb predicate are filled with the arguments of their indirect and prepositional objects. In the example presented above the predicate *eat* has arguments after *;* (semicolon) which indicates its adjuncts. The distinction between complements and adjuncts is a novelty to the LF proposed by the authors in this paper. For this first trial, we did not make the distinction between the two and thus the accepted representation would be `eat:v_(e2, x3, x2, x4)` - see below.

Output: `Earth:n_(x1) provide:v_(e1, x1, x2) food:n_(x2) we(x3) eat:v_(e2, x3, x2, x4) day:n_(x4)`

Predicates are formed by the concatenation of the base form of the word and its lexical category as encoded in WordNet (since only *nouns, verbs, adjectives and adverbs* are encoded in WordNet, only predicates for those lexical categories have the category attached to the predicate).

To ease the task, the notation was relaxed by adopting few simplifications similar, to some extent, to the simplications in (Moldovan and Rus, 2001): determiners, plurals, negation, auxiliaries and verb tenses, punctuation are ingnored. Collocations, such as *New York*, should be considered a single predicate as well as verbs having particles (e.g. *give up*). For cases when an argument is underspecified, such as the logical subject in *Jim was told to*

say something, an artificial argument should be generated.

The advantages of the LF notation are mantfold:

- it allows a simple syntax/semantics interface
- it is user friendly
- it has positional syntactic arguments that ease other NLP tasks such as textual interpretationa and textual inference
- if predicates are disambiguated with respect to a general ontology such as WordNet it leads to *concept predicates*
- it is easily customizable (for example to distinguish between arguments and adjuncts)

For details about the principles of Logic Forms read Chapter 2 in (Rus, 2002), (Rus and Moldovan, 2002) and (Hobbs, 1986). The LF notation proposed for the LFi competition is novel, and different from the one described in the previous references since it distinguishes between complements and adjuncts among other differences. A web page for the LFi task is available at *http://www.cs.iusb.edu/ vasile/logic/indexLF.html* and a discussion group, called *logicform*, was opened at *yahoo.groups.com* which can also be consulted.

3 Test Data

The test data was compiled so that the impact of external tools that different sytems might use in the LF identification process be minimal. For example, it is well-known that the accuracy of automatic syntactic parsing drops drastically for sentences larger than 40 words and thus we kept the size of the collected sentences below the 40 words threshold. The average sentence size in the test data is 9.89 words.

Special attention was paid to covering linguistic phenomena such as: coordination, compound nouns, ditransitives, multiword expressions (give up, as well as, etc.), relative clauses and others. Different sources were used to look up such cases: Treebank, WordNet and the web.

The size of the test set (4155 arguments, 2398 predicates, 300 sentences) allows a better evaluation of the vertical scalability (coverage of as many linguistics problems as possible) of sytems rather than their horizontal scalability (handling large data sets without significant deterioration of performance displayed on small sets).

4 Annotation Guidelines

The annotation part is the most critical part of any evaluation exercise. For the Logic Form Identification task the following steps were applied to obtain the correct LF for the test data:

1. logic forms for the test data were automatically obtained using an extended version of the LF derivation engine developed in (Rus, 2002) for LFi of WordNet glosses. As part of this step, sentences were preprocessed: tokenized (separating punctuation from words) using the Penn Treebank guidelines, tagged with Brill's tagger (Brill, 1992) and then parsed with Collins' statistical parser (Collins, 1996).
2. a first manual checking of the previously generated LF was done.
3. a second manual checking was done by another annotator.
4. quality assurance of the previous steps was performed by individual annotators by checking specific cases (ditransitives, relative pronouns, etc.) with much emphasis on consistency.
5. annotators agreement was done with a human moderator solving conflicting cases.

5 Metrics

Two performance measures to evaluate Logic Form Identification methods were developed by Rus in (Rus and Moldovan, 2002) for the particular task of LFi for WordNet glosses (the definitions of concepts are shorter than regular sentences in terms of number of words, etc.). Each measure has advantages in some context.

Predicate level performance is defined as the number of predicates with correct arguments divided by the total number of predicates. This measure focuses on the derivation method, though at a coarse-grained level because it does not capture the capability of a method to successfully identify a specific argument, e.g. *the subject of a verb*.

Gloss level performance is the number of entire glosses correctly transformed into logic forms divided by the total number of glosses attempted. This measure catches contextual capabilities of a method in that it gives an idea of how well a method performs at gloss level. It is a more appropriate measure when one tries to see the impact of using full glosses in logic forms to applications such as planning. This measure is specific to the particular task of LFi for concept definitions and thus is not suited for general open text tasks.

Let us consider the following gloss from WordNet:

Abbey is a convent ruled by an abbess.

and let us suppose that some system, say *Sys* is able to generate the following logic form (please note that the subject of *rule* event is missing):

```
abbey(x1) & be(e1, x1, x2) &
convent(x2) & rule(e2, _ , x2) &
by(e2, x3) & abbess(x3)
```

Since one of the arguments is missing the predicate level performance is 5/6 (there are 6 predicates and for five of them the system generated all the arguments correctly) and the gloss level performance is 0/1 (this measure awards cases where all the predicates in the statement have all their arguments correctly assigned).

None of the two measures can distinguish between two systems, where one misses the subject of the *rule* event and the other misses both the subject and object (both systems will miss one predicate).

We propose two new, finer metrics in the next section, that are more suitable for a less restrictive LFi task: precision and recall. Both precision and recall can be defined at argument and predicate level, respectively.

5.1 Argument Level

We define *Precision* at argument level as the number of correctly identified arguments divided by the number of all identified arguments. *Recall* at argument level is the number of correctly identified arguments divided by the number of arguments that were supposed to be identified.

5.2 Predicate Level

Precision at predicate level is the number of correctly and fully identified predicates (with ALL arguments correctly identified) divided by the number of all attempted predicates. *Recall* at predicate level is the number of correctly and fully identified predicates (with ALL arguments correctly identified) divided by the number of all predicates that were supposed to be identified.

Let us suppose that some system outputs the following logic form for the above example:

Sample Output: `Earth:n_(x1)`
`provide:v_(e1, x1, x2) food:n_(x2)`
`we(x3) eat:v_(e2, x3, x4)`
`day:n_(x4)`
Correct Output: `Earth:n_(x1)`
`provide:v_(e1, x1, x2) food:n_(x2)`
`we(x3) eat:v_(e2, x3, x2, x4)`
`day:n_(x4)`

where *x4* is incorrectly indentified as the direct object of *eating* event. In the correct output there are 11 slots to be filled and the predicate *eat* should have 4 arguments. The previously defined measures for the sample output are given in Table 1:

Metric / Level	Argument	Predicate
Precision	9/10	5/6
Recall	9/11	5/6

Table 1: Examples of *Precision* and *Recall* at argument and predicate level.

In addition, we report a more global measure called *exact sentence* which is defined as the number of sentences whose logic form was fully identified (all predicates and arguments correctly found) divided by the number of sentences attempted. This is similar to gloss level performance measure presented before. We proposed and computed several variants for it which are described below.

Sentence-Argument (Sent-A): How many sentences have ALL arguments correctly detected out of all attempted sentences.

Sentence-Predicate (Sent-P): How many sentences have ALL predicates correctly detected out of all attempted sentences.

Sentence-Argument-Predicate Sent-AP: How many sentences have ALL arguments correctly detected out of sentences which have ALL predicates correctly detected

Sentence-Argument-Predicate-Sentences Sent-APSent: How many sentences have ALL arguments and ALL predicates correctly detected out of all attempted sentences.

6 Extra Resources

A package of trial data was provided to interested participants. The trial package contains two data files: (1) English sentences and (2) their corresponding logic form. A software evaluator was available for download on the web page of the task. We compiled a dictionary of collocations from WordNet which was also freely available for download. It includes 62,611 collocations.

7 Submission Format

Each team was supposed to submit a file containing on each line the answer to a input sentence using the following pattern:

InstitutionShortName Y000 Sentence# Score :: Logic Form

Here is an example:

	Argument Level		Predicate Level	
Team / *Metric*	*Precision*	*Recall*	*Precision*	*Recall*
University of Amsterdam (ams)	0.729	0.691	0.819	0.783
Language Computer Corporation (lcc)	0.776	0.777	0.876	0.908
MITRE (mitre)	0.734	0.659	0.839	0.781
University of Sydney (syd)	0.763	0.655	0.839	0.849
	Sent-A	*Sent-P*	*Sent-AP*	*Sent-APSent*
University of Amsterdam (ams)	0.256	0.320	0.510	0.163
Language Computer Corporation (lcc)	0.236	0.516	0.419	0.216
MITRE (mitre)	0.266	0.213	0.406	0.086
University of Sydney (syd)	0.160	0.353	0.386	0.136

Table 2: Comparative view of valid submissions.

IUSB Y000 3 89.7 :: nn:_ (x1, x2, x3) logic:n_ (x2) form:n_ (x3)

The field Y000 was generated as is, for all lines. It will be used in future trials.

8 Results

Initially, there were 27 teams registered to participate in the Logic Form Identification task and 6 submissions were received by the deadline. One of the submissions was discarded since the file contained no valid data and another one was not included in the comparative results shown in Table 2 since it used manual parsing (parsing is not a necessary step in obtaining the LF). The part of speech info attached to some predicates was ignored when computing the scores. We plan to use it in further trials.

If one looks at the results in the table one may notice their consistency. At Argument level precision and recall range from 0.729 to 0.776 and from 0.655 to 0.777, respectively. The same trend can be observed at Predicate level (the results are slighlty better). At a more coarse-grain level (Sentence level) the results vary more but still one can distinguish a certain degree of consistency: the *Sent-A* measure ranges from 0.160 to 0.256 and the *Sent-AP* measure varies from 0.386 to 0.510.

9 Conclusion

In the first attempt to systematically evaluate LFi systems we managed to provide a gold standard, a first software evaluator and proposed few performance metrics that can be used by researchers in the community to further study their approaches.

Among the drawbacks, it is worth mentioning the lack of training data which we plan to offer in the future.

The results reported by different systems constitute a lower bound of their approaches since the test data comprised raw sentences and thus the reported performances include errors coming from tokenization, part of speech tagging and parsing, wherever parsing was used.

Due to a tight schedule it was not possible to analyze the different approaches adopted by different systems but we hope the ACL meeting will provide the necessary background information and discussions to foster the development of such a study.

References

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 152-155.

Michael John Collins. 1996. A new statistical parser based on bigram lexical dependencies. In Arivind Joshi and Martha Palmer, editors, *Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics*, pages 184–191, San Francisco. Morgan Kaufmann Publishers.

Jerry R. Hobbs. 1986. Overview of the TACITUS project. *Computational Linguistics*, 12(3).

Dan I. Moldovan and Vasile Rus. 2001. Logic Form transformation of wordNet and its Applicability to question answering. In *Proceedings of ACL 2001*, Toulouse, France, 6-11 July. Association for Computational Linguistics.

Vasile Rus and Dan Moldovan. 2002. High precision logic form transformation. In *International Journal for Tools with Artificial Intelligence*. IEEE Computer Society, IEEE Press, September.

Vasile Rus. 2002. *Logic Form for WordNet Glosses and Applications*. Phd thesis, Southern Methodist University, May.