interspeech 2008 incorporating SST 08

A Methodology and Tool Suite for Evaluation of Accuracy of Interoperating Statistical Natural Language Processing Engines

Uma Murthy[1], *John F. Pitrelli*[2], *Ganesh Ramaswamy*[2], *Martin Franz*[2] *and Burn L. Lewis*[2]

[1]Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA
[2]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
umurthy@vt.edu, {pitrelli, ganeshr, franzm, burn}@us.ibm.com

10.21437/Interspeech.2008-536

Abstract

Evaluation of accuracy of natural language processing (NLP) engines plays an important role in their development and improvement. Such evaluation usually takes place at a per-engine level. For example, there are evaluation methods for engines such as speech recognition, machine translation, story boundary detection, etc. Many real-world applications require combinations of these functions. This has become possible now with NLP engines attaining sufficient accuracy to be able to combine them for complex tasks. However, it is not evident how the accuracy of output of such aggregates of engines will be evaluated. We present an evaluation methodology to address this problem. The key contribution of our work is an extensible methodology that narrows down possible combinations of machine outputs and ground truths to be compared at various stages in an aggregate of interoperating engines. We also describe two example evaluation modules that we developed following this methodology.

Index Terms: evaluation, accuracy, natural language processing, NLP, interoperation, UIMA

1. Introduction

Evaluation of accuracy of output[1] of natural language processing (NLP) engines helps in the development and improvement of these engines. Typically, evaluation of an engine involves comparing the output of the engine with a *ground truth*, a reference output (typically, human-generated) for the same task that is performed by the engine. Closeness to the ground truth indicates a high quality processing engine. For example, in the evaluation of a machine translation engine, the output of the engine is compared with a human-generated translation. Manual evaluation of the engine can be expensive and time consuming, especially if it needs to be done often in building and refining the engine. This has motivated the development of automatic evaluation techniques such as the BLEU score [1]. Another example of automatic evaluation is of machine-detected story boundaries using the TDT sliding-window-style evaluation [2].

Many real-world applications require combinations of NLP functions. For example, to convert speech in a source language to speech in a target language, one could use a succession of NLP functions – speech recognition, machine translation, and text-to-speech synthesis. In such cases, automatic evaluation becomes even more challenging because, instead of a single engine, there is an aggregate of interoperating engines. For NLP systems with many components, previous work on NLP evaluation mentions two kinds of evaluations: *Intrinsic* – how a particular component works in its own terms, and *Extrinsic* – how it contributes to the overall performance of the system [3]. Yet, most work on NLP accuracy evaluation has focused on evaluation of a single NLP engine at a time. In the case of single-engine evaluation, the goal has been to define a precise set of evaluation criteria, which may include evaluation test data and evaluation metrics. We see such standard evaluation and data sets with regard to information retrieval [4], topic detection and tracking [2, 5], speech recognition (word-error rate), etc. However, to the best of our knowledge, there have not been any efforts in evaluating large aggregates of interoperating engines.

To address this problem, we have developed an evaluation methodology that systematically narrows down possible combinations of machine outputs and ground-truths to be compared at various stages in an aggregate. The main issues to consider here are: 1) How is the accuracy of one engine or a set of engines evaluated, in the context of being present in an aggregate?; 2) What is the measure of accuracy of an aggregate and how can it be computed?; and 3) How can the mechanics of this evaluation methodology be validated and tested?

Since this area of evaluating aggregates of engines is fairly new, we are following an iterative approach to address this problem. The methodology presented in this paper is a starting point. Following this methodology, we implemented example evaluation modules for two small aggregates. Preliminary testing of these modules indicates that the methodology is promising. Note that the focus here is to validate the mechanics of evaluation rather than the engines themselves. Further testing and development of more evaluation modules will make this methodology and tool suite a more exhaustive and robust template for evaluation of aggregates of interoperating NLP engines.

2. Background and Context

This research is in conjunction with the GALE Interoperability Demonstration system (IOD) project [6], which is built upon Unstructured Information Management Architecture (UIMA) [7, 8]. UIMA facilitates interoperation of sets of distributed engines employing heterogeneous computing environments for analyzing less-structured information (text, audio, video, images, etc.). Each analysis engine annotates various parts of the input to provide meaning and structure, which may later be processed more easily. For example, a translation engine annotates one language's text by associating its translation with it.

IOD is a system of distributed interoperating NLP engines [6], consisting of transcription, translation and information extraction functions. The engines in IOD perform the following

[1]In this work, the term evaluation refers to evaluation of accuracy of output; we do not explore other evaluation metrics such as speed and memory usage.

functions:

- speech recognition (speech-to-text or "STT"),
- entity detection ("ED"),
- machine translation ("MT"),
- multi-engine MT ("MEMT"),
- story-boundary detection ("SBD"),
- topic clustering of stories ("TC"),
- multi-story topic summarization ("TS"), and
- headline generation ("HG").

IOD processing begins with Arabic audio segments (extracted from Arabic news videos) being input into the STT engine, which transcribes the audio into Arabic text. This text may then be translated to English text using MT engines. On both the Arabic text and the English text, several information extraction tasks may be performed such as ED, TC, TS and HG. Finally, the MEMT engine processes the results from various combinations of the preceding engines and assembles a translation which tends to be better than that from any of the single STT-MT engine combinations.

One of the goals of IOD is to help users configure and run their own aggregates of engines. To determine how well these aggregates are functioning, users must be able to evaluate the accuracy of the engines' combined output. It is important for the evaluation to be automatic since a user may not have specific domain knowledge about a particular engine and may only be interested in some indicators that would facilitate him/her to make decisions about selection of best possible engines to use for the task at hand. The evaluation should be quick and easy, hence automatic. The methodology described in this paper is a step towards achieving this automatic evaluation.

3. Evaluation Methodology

The core of the evaluation methodology is the *evaluation space* for machine and human-generated outputs that would be compared at each stage in an aggregate. By evaluation space, we mean the pattern of human- and machine-generated labeling used to generate a reference against which the aggregate's output is to be compared. In order to do that, we took an example aggregate and listed all possible combinations of machine and human-generated outputs at every stage in the aggregate. Then, we narrowed down this list based on what made for a meaningful comparison at each stage in the aggregate and also so that this pattern would be extensible for aggregates of varying sizes and complexity.

For example, consider a small aggregate with two engines, $\mathrm{STT} \rightarrow \mathrm{SBD}$, as shown in Figure 1. In the figure, M stands for machine output and H stands for human-generated output. The top part of the figure shows the engines in the aggregate and the bottom part shows the various ground truth options at every stage in the aggregate. Straight, bold lines represent processing by engines or humans and the dotted lines indicate the possible evaluation pairs. If we were to evaluate just STT, we would compare the machine output of STT with the ground truth, which is the human-generated output. When we add SBD to this aggregate, we have two ground truths that may be compared to the machine output:

- M-H: In this case, the ground truth is human generated story boundaries on an STT transcript. Comparing this with the aggregate output, M-M, will give us an indication of the accuracy of SBD. This is different from evaluating SBD in a stand-alone mode. Here, we are evaluating SBD in the context of it processing the output of a specific STT engine. Thus, this evaluation will give us an idea of errors generated by SBD in the context of an STT-SBD aggregate.
- H-H: Comparing this with the aggregate output, M-M, will provide an indication of the errors generated as a result of the combined processing of STT+SBD.

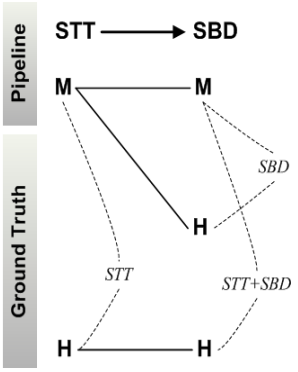

Figure 1: *Evaluation space for an aggregate with two engines – STT and SBD.*

Figure 1 shows the evaluation space for an aggregate of two engines. The complexity of the evaluation space increases significantly when more engines are added to the aggregate. We see this in Figure 2, where the engines are: $\mathrm{STT} \rightarrow \mathrm{SBD} \rightarrow \mathrm{TC} \rightarrow \mathrm{TS} \rightarrow \mathrm{HG}$. There are a few things to note here:

- Comparison of the output of an aggregate (M-M-M...) with a completely human-generated output (H-H-H...) evaluates the performance of the entire aggregate.
- We do not consider a ground truth case that includes an H-M sequence (such as H-M, M-H-M, etc.). We are not completely ruling out this possibility. Instead, we have chosen not to focus on such comparisons for now, as we believe that they may have less-direct interpretations.
- A pattern emerges in the ground truth as one progresses upward from a completely human-generated output toward a ground truth with only one H component (one row short of a completely machine-generated output). The comparison of the ground truth and the machine output results in the evaluation of the combination of the rightmost engines, in the context of processing by the preceding engines. For example, for the five-engine aggregate as shown in Figure 2, by comparison with the machine output represented by the top row, the lowest row's ground truth is used for evaluation of the combined performance of STT+SBD+TC+TS+HG, the next-higher row's ground truth provides for the evaluation of the combined performance of SBD+TC+TS+HG in the context of post-processing STT, and so on – finally ending in the evaluation of just HG, in the context of post-processing the first four engines.
- Comparing two vertically adjacent evaluations, such as TS+HG and TC+TS+HG, represents measuring how much one engine (here, TC) degrades the accuracy of an aggregate.

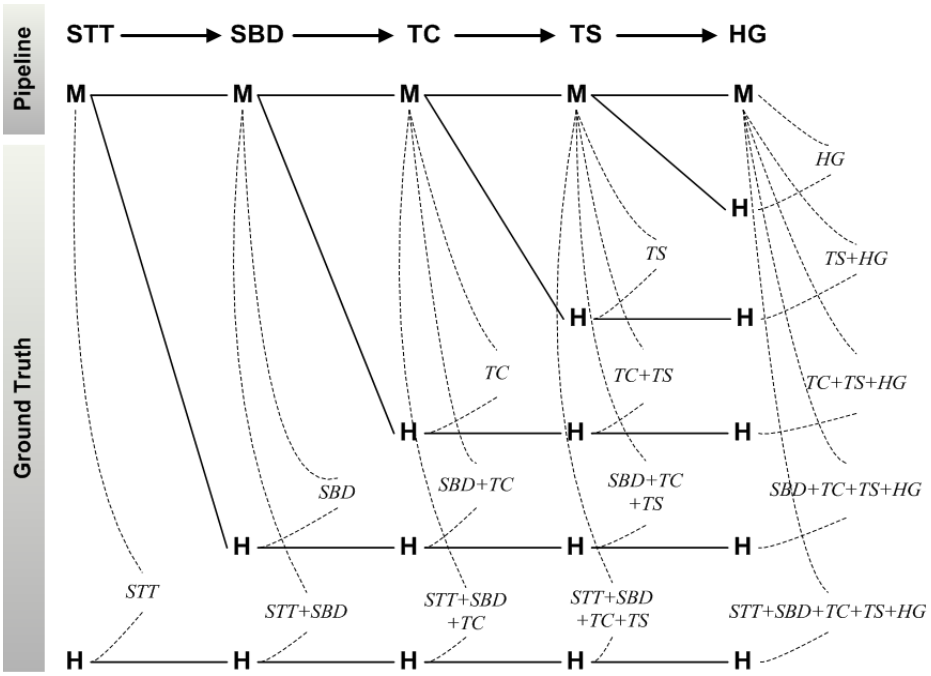

Figure 2: *The evaluation space for an aggregate with five engines STT, SBD, TC, TS, and HG.*

- Any point in this evaluation space can be used to compare two engines for the same function. For example, competing STTs may be evaluated by performing any evaluation of M1-M-... against any appropriate ground truth, then repeating with M2-M-... against the same ground truth.

- The biggest challenge of such an array of evaluations is the labor/expense of assembling various ground truths at different stages.

Using this evaluation space as a guide, we developed two example evaluation modules. The function of an evaluation module is to evaluate all combinations of ground truths and machine outputs for an aggregate as specified by the evaluation space. Note that for a five-engine aggregate (as shown in Figure 2), comprehensive evaluation would require a suite of modules that covers all 15 evaluation points. However, the mechanics of evaluation may remain the same for different comparisons and only the ground truth varies[2]. First, we explored methods that are used to evaluate the last engine in the aggregate. This is because the output of an aggregate will usually be in the format of the output of the last engine in the aggregate (as when that engine is functioning in a stand-alone mode). The next step was to modify this evaluation method based on the (input and output of) engines upstream in the aggregate.

Validation and testing of an evaluation module varied depending on: 1) the evaluation module; 2) the ground truth formats being considered; and 3) the engines in the aggregate. Further, testing a module with as many test data sets and ground truths (of all formats) as possible, would strengthen that module.

[2]The mechanics may sometimes vary depending on the varied formats of ground truths.

4. Example Evaluation Modules

We developed evaluation modules for two aggregates. The focus of testing a module was on validating the mechanics of the evaluation rather than validating anything relating to the data or engines in question.

4.1. STT–SBD

The evaluation module for the STT–SBD aggregate was based on evaluation methods for SBD engines (the last engine in this aggregate). Most currently-used story-segmentation measures, such as TDT Cseg [9], are based on counting the number of missed and incorrectly-proposed boundaries. The counts are smoothed by a sliding-window scheme to allow partial credit for boundaries at incorrect positions that are close to the reference boundaries. These values are computed for various threshold levels on the confidence scores output by the SBD engine. This evaluation produces an ROC curve and can compare two such curves at an operating point, equal-cost/error points, etc.

We tested this evaluation module with data from six Arabic news shows, where each show consisted of 30 two-minute audio segments. The only ground truth available to us was of type M'-H (similar to M-H with the M output being from a different STT engine). We used this as an approximation of the M-H ground truth type. Through this test, we were automatically able to obtain ROC curves for the above-mentioned test data.

4.2. STT–MT

We based the evaluation method for the STT–MT aggregate on the BLEU metric [1]. BLEU is an automatic MT evaluation method that highly correlates with human judgment of translation quality and has marginal cost per run. The central idea behind the BLEU method is that "the closer a machine translation is to a professional human translation, the better it is". The BLEU score is a measure of this closeness and is fashioned after

the word-error-rate metric. It is computed by taking a weighted average of variable-length-phrase (N-gram) matches against a reference translation. The BLEU score ranges from zero to one. For the STT–MT evaluation module, this metric is appropriate since the output format would always be a target language translation for all types of ground truths for this aggregate.

We tested the evaluation module on a data set of 19 audio segments, of an average 2 minutes each . The ground truth on this data was a human-generated output from the speech segments. Since the ground truth was generated directly from the speech, this would correspond to the H-H ground truth type in the evaluation space. We were able to test this module with two MT engines – from IBM [10] and RWTH Aachen [11]. The output of this test was that we were able to automatically obtain BLEU scores for two interchangeable engines from two different sites.

5. Discussion and Future Work

We presented initial work on a methodology for evaluating an aggregate of interoperating NLP engines, as well as two example evaluation modules. Preliminary testing of these modules indicate that the methodology is promising, in that they help validate the mechanics of the evaluation method to some extent. The most challenging aspect of the project was the lack of availability of different ground truth types, as specified in the evaluation space. Obtaining ground truth data can be costly. Since the modules that we developed were for aggregates with a small number of engines, we were able to get this type of ground truth data, or derive approximations of the available reference outputs. This can become more challenging for large engine-aggregates.

Our immediate next step with regard to the evaluation modules developed, would be to run more rigorous tests on them. Specifically, we would like to test with the following conditions:

- Test with actual ground truth data: Instead of using approximations, it would be ideal to get actual ground truth data.
- Test with more data sets: Testing with different test data will help us gauge the accuracy consistency of the evaluation module.
- Test on different engines of the same kind and varying-quality (human or machine) outputs: This kind of test will allow us to see if the evaluation module is able to distinguish between good and poor quality machine output. It will also allow us to compare the accuracy of different engines of the same kind, as we did in the case of the STT–MT.

With regard to the evaluation methodology, we believe that we have only scratched the surface of a vast, and relatively unexplored, area of research. Some of the goals of an evaluation methodology for evaluating an aggregate of engines may be:

1. Compute accuracy of output of an aggregate of engines – how close is the output to some ground truth?
2. Identify where in the aggregate an error appears – which engines in the aggregate are causing erroneous aggregate output?
3. Determine what the measurable contribution of an individual engine is on an erroneous aggregate output; and later, how much each engine needs to be "tuned" or "normalized" to get optimal aggregate output.

Through the modules developed, we have been able to achieve goal 1 to some extent. We believe that our evaluation methodology can be used and extended to achieve goal 2 and goal 3, although there is a lot of work to be done in both of these areas. Also, future work will include development of more evaluation modules for larger and more complex aggregates, performing direct comparisons within columns of Figure 2, and reuse of modules and their combinations for evaluating other aggregates.

6. Acknowledgements

Many thanks to Yaser Al-Onaizan, Jason Pelecanos, Pierre Dognin, Abe Ittycheriah, Janice Kim, Mohamed Nasr, Jason Pelecanos, Leiming Qian, George Saon, Paola Virga, Lexing Xie and Jian-Ming Xu for their valuable support, ideas and feedback. This work was supported in part by DARPA under contract HR0011-06-2-0001.

7. References

[1] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[2] J. Allan, Ed., *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publishers, 2002.

[3] K. S. Jones and J. R. Galliers, *Evaluating Natural Language Processing Systems: An Analysis and Review*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.

[4] Text retrieval conference (from NIST). [Online]. Available: http://trec.nist.gov

[5] "The 2002 topic detection and tracking (TDT 2002) task definition and evaluation specification version 1.1 from NIST," 2002. [Online]. Available: http://www.nist.gov/speech/tests/tdt/tdt2002/evalplan.htm

[6] J. F. Pitrelli, B. L. Lewis, E. A. Epstein, M. Franz, D. Kiecza, J. L. Quinn, G. Ramaswamy, A. Srivastava, and P. Virga, "Aggregating Distributed STT, MT, and Information Extraction Engines: The GALE Interoperability-Demo System," in *INTERSPEECH 2008*, 2008.

[7] D. Ferrucci and A. Lally, "UIMA: an architectural approach to unstructured information processing in the corporate research environment," *Nat. Lang. Eng.*, vol. 10, no. 3-4, pp. 327–348, 2004.

[8] "Apache UIMA," 2002. [Online]. Available: http://incubator.apache.org/uima/index.html

[9] G. Doddington, "The topic detection and tracking phase 2 (TDT2) evaluation plan," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 223–229.

[10] Y. Al-Onaizan and K. Papineni, "Distortion models for statistical machine translation," in *ACL '06: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, 2006, pp. 529–536.

[11] O. Bender, E. Matusov, S. Hahn, S. Hasan, S. Khadivi, and H. Ney, "The RWTH Arabic-to-English spoken language translation system," *IEEE Workshop on Automatic Speech Recognition & Understanding, 2007. ASRU.*, pp. 396–401, 9-13 Dec. 2007.