Comment-guided Learning: Bridging the Knowledge Gap between Expert Assessor and Feature Engineer

Xiang Li, Wen-Pin Lin, Heng Ji
Computer Science Department, Queens College and Graduate Center
City University of New York
New York, USA
{jackieiuu729,danniellin,hengjicuny}@gmail.com

***Abstract*—As more and more natural language processing systems utilize human assessment on system responses, it becomes beneficial to discover some hidden privileged knowledge (such as comments) from assessors. We present a simple, low-cost but effective comment-guided learning approach to exploit such knowledge declaratively in an automatic assessor. Our approach only requires a small set of training data, together with comments which are naturally available from human assessment. To demonstrate the power and generality of this approach, we apply the method in two very different applications: name translation and residence slot filling. Our approach achieved significant absolute improvement (15% for name translation and 8% for slot filling) over state-of-the-art systems. It also outperformed previous methods such as Recognizing Textual Entailment (RTE) based fact validation. Furthermore, it can be used as feedback to significantly speed up human assessment.**

Keywords-comment-guided learning; assessment; feature engineering.

I. Introduction

As an inter-disciplinary area, statistical Natural Language Processing (NLP) requires two crucial aspects: (1) good choice of machine learning algorithms; (2) good feature engineering. For many NLP tasks, feature engineering significantly affects the performance of systems. However, feature engineering is very challenging because it encompasses feature design, feature selection, feature induction and studies of feature impact, all of which are very time-consuming, especially when there are a lot of data or errors to analyze. As a result, in a typical feature engineering process, the system developer is only able to select a representative data set as the development set and analyze partial errors.

On the other hand, recently many NLP tasks have moved from processing hundreds of documents to large-scale or even web-scale data. Once the collection grows beyond a certain size, it is not feasible to prepare a comprehensive answer key in advance. Because of the difficulty in finding information from a large corpus, any manually-prepared key is likely to be quite incomplete. Instead, we can pool the responses from various systems and have human assessors manually review and judge the responses. Assessing pooled system responses as opposed to identifying correct answers from scratch has provided a promising way to generate training data for NLP systems.

However, almost all of the previous NLP systems only utilized the direct assessment results (correct, incorrect, etc.) for training, while ignoring the valuable knowledge hidden in the human assessment procedure. If we consider an NLP system as a "student" while the human assessor as a "teacher", then the "homework grades" (i.e., assessment results) are just a small part of the teacher's role. Besides grading, a teacher also provides explanations about why an answer is wrong, comments about what kind of further knowledge the student can benefit from, and so on. Similarly, when a human assessor makes a judgement, he/she must know the reasons to verify it. As a result, it will cost little extra time for an assessor to write down their comments, because the comments can be naturally derived from a small yet representative sample data set that the assessor has judged. Such assessment results and comments are not available in the test phase. However, since a system tends to make similar types of errors on various data sets, it can always benefit from such comments for new runs.

In this paper, we propose a new and general Comment-Guided Learning (CGL) framework in order to fill in the gap between the expert annotator and the feature engineer (Section 3). This framework aims to encode features with the guidance of comments from human assessors in a re-scoring step. In order to verify the efficacy of this approach, we shall conduct case studies on two distinct application domains: a relatively simple name translation task (Section 4) and a more challenging residence slot filling task (Section 5). Empirical studies demonstrate that with about little longer annotation time, we can significantly improve the performance for both tasks.

II. Related Work

Vapnik [1] proposed to incorporate more of "teacher's role" (i.e., privileged knowledge) into traditional machine learning paradigm, and pointed out that such privileged knowledge may not be available during the test phase. We follow this basic idea and incorporate additional feedback from the comments into assessment, so that we can still utilize the teacher's role as the final step of the test phase. Vapnik's work aimed to improve a classifier using privileged knowledge by exploiting information from a different classification space to tune the classifier to make better predictions. Their updated classifier is still applied to the same space as the original one. In contrast, we encode the comments from assessors as new features for an automatic assessor. The updated system consisted of the baseline and the assessor is applied to a new classification space (i.e., new test instances).

Castro et al. [2] investigated a series of human active learning experiments. Our experiment of using CGL to speed up human assessment exploited assistance from multiple systems.

Our idea of learning from error corrections is also similar to Transformation-based Error-Driven Learning, which has been successfully applied in many NLP tasks such as part-of-speech tagging [3] and semantic role labeling [4]. In these applications the transformation rules are automatically learned based on sentence contexts at each iteration. However, our applications require global knowledge which may be derived from diverse linguistic levels and vary from one system to the other, and thus it's not straightforward to design and encode transformation templates. Therefore in this paper we choose a more modest way of exploiting the comments encoded by human assessors.

Most of the previous name translation work combined supervised phonetic similarity based name transliteration approaches with Language Model based re-scoring (e.g., [5]). But none of these approaches exploited the feedback from human assessors. There are many other alternative automatic assessment approaches for slot filling. Besides the RTE-KBP validation [6] discussed in the paper, some slot filling systems also conducted filtering and cross-slot reasoning (e.g., [7]; [8]) to improve results.

III. Comment-guided Learning

A. General Framework

In Table I, we aim at formalizing the mapping of some essential elements in human learning and machine learning for NLP. We can see that among these elements, little study has been conducted on incorporating the comments made by human assessors. In most cases it was not the obligation for the human assessors to write down their comments during assessment. In contrast, the human learning scenario involves more interactions. However, we can assume that any assessor is able to verify and comment on his/her judgement. Based on this intuition we propose a new comment-guided learning (CGL) paradigm as shown in Figure 1. This algorithm aims to extensively incorporate all comments from an old development data set (i.e., "old homework" in human learning) into an automatic correction component. This assessor can be applied to improve the results for a new test data set (i.e., "new homework" in human learning).

Table I
Some Elements in Human Learning and Machine Learning for NLP

Human Learning	Machine Learning for NLP	Examples of existing NLP approaches
student	system	baseline system
teaching assistant	human annotator	
teacher	human assessor	
lectures	training data	
graded home-work	assessed system output	
graded home-work with comments	assessed system output with comments	None
erroneous homework set	negative samples/errors	transformation based learning
homework re-view against lecture	system output with background documents	recognizing textual entail-ment
group study	pooled system responses	voting, learn-ing-to-rank

B. Detailed Implementation

The detailed CGL algorithm can be summarized as follows.

1. The pipeline starts from running the baseline system to generate results. In this step we can also add the outputs from other systems (i.e., classmates in human learning) or even human annotators (i.e., Teaching Assistant (TA) in human learning). We will present one case study on slot filling, which incorporates these two additional elements, and the other case study on name translation which only utilizes results from the baseline system.

2. We obtain comments from human assessors on a small development set D^i. Each time we ask a human assessor to pick up N (N=3 in this paper, the value of 3 was arbitrarily chosen; various in this number of clusters produce only small changes in performance.) random results to generate one new comment. One could impose some pre-defined format or template restrictions for the comments, such as marking the indicative words as rich annotations and encoding them as features. However, we found that most of the expert comments are rather implicit and even requires global knowledge. Nonetheless, these comments represent general solutions to reduce the common errors from the baseline system.

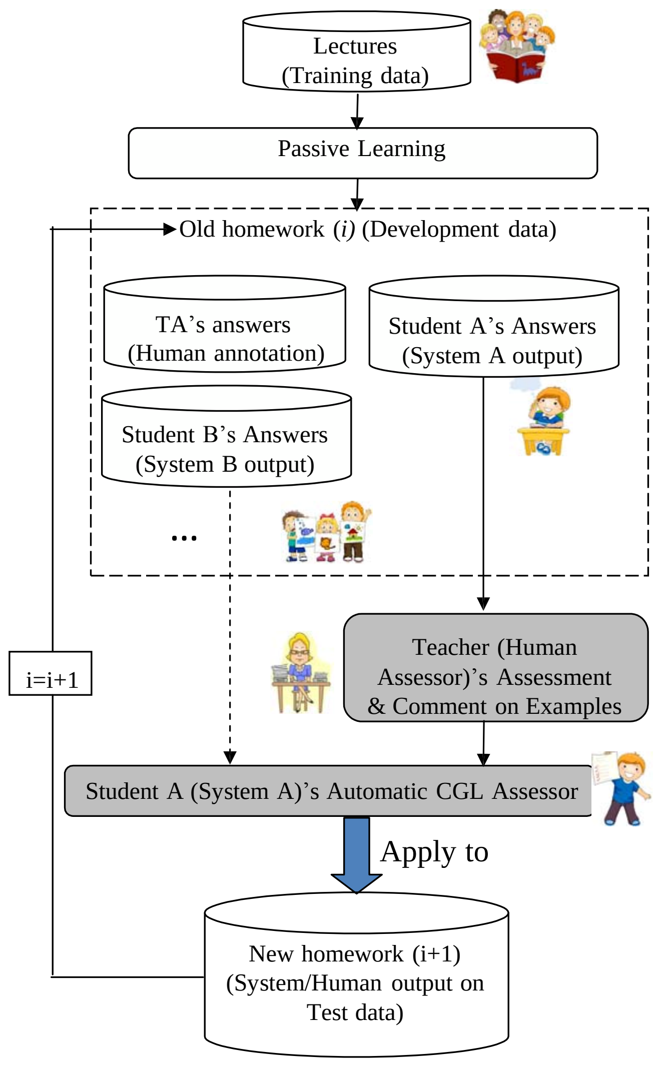

Figure 1. Training a CGL Assessor

3. We encode these comments into features. We then further train a Maximum Entropy (MaxEnt) based automatic assessor A^i using these features. For each response generated from the baseline system, A^i can classify it as correct or incorrect. We choose a statistical model instead of rules because heuristic rules may overfit a small sample set and highly dependent on the order. In contrast, MaxEnt model has the power of incorporating all comments into a uniform model by assigning weights automatically. In this way we can integrate assessment results tightly with comments during MaxEnt model training.

4. Finally, A^i is applied as a post-processing step to any new data set D^{i+1}, and filter out those results judged as incorrect.

The algorithm can be conducted in an iterative fashion. For example, human assessors can continue to judge and provide comments for D^{i+1} and we can update the automatic assessor to A^{i+1} and apply it to a new data set D^{i+2}, and so on.

IV. Name Translation Mining

This section presents the first case study of applying CGL for name translation validation.

A. Task Definition and Baseline System

Name translation is important well beyond the relative frequency of names in a text: a correctly translated passage, but with the wrong name, may lose most of its value. Some recent work explored unsupervised or weakly-supervised name translation mining from large-scale data. For example, Bouma et al. [9] aligned attributes in Wikipedia Infoboxes based on cross-page links; Lin et al. [10] described a parenthesis translation mining method; You et al. [11] applied graph alignment algorithm to obtain name translation pairs based on co-occurrence statistics. However, these approaches suffer from low accuracy and thus it is important to develop automatic methods to evaluate whether the mined name pairs are correct or not. In this paper we focus on validating person name translations by encoding the comments which human assessors made on a small data set. We applied an unsupervised learning approach as described in [12] as our baseline system, to mine name translation pairs from English and Chinese Wikipedia Infoboxes.

B. Comments and Feature Encoding

The detailed comments used for validating name translations are as follows.

- Comment 1: "these two names do not co-occur often"
  This comment indicates that we can exploit global statistics to filter out some obvious errors, such as "Ethel Portnoy" and "Chen Yao Zu". Using Yahoo! search engine, we compute the co-occurrence, conditional probability and mutual information of a Chinese Name CHName and an English name ENName appearing in the same document from web-scale data with setting a threshold for each criteria.
- Comment 2: "these two names have very different pronunciations"
  Many foreign names are transliterated from their origin pronunciations. As a result, person name pairs (e.g., "Lomana LuaLua" and "Luo Ma Na . Lu A Lu A") usually share similar pronunciations. In order to address this comment, we define an additional feature based on the Damerau-Levenshtein edit distance ([13]) between the Pinyin form of CHName and ENName. Using this feature we can filter out many incorrect pairs, such as "Maurice Dupras" and "Zhuo Ya . Ke Si Mo Jie Mi Yang Si Qia Ya".
- Comment 3: "these two names have different profiles"
  When human assessors evaluate the name translation pairs, they often exploit their world knowledge. For example, they can quickly judge "Comerford Walker" is not

a correct translation for "Sen Gang Er Lang (Jiro Oka Mori)" because they have different nationalities (one is U.S. while the other is Japan). To address this comment, we define the profile of a name as a list of attributes. Besides using all of the Wikipedia Infobox values, we also run a bi-lingual information extraction (IE) system [14] on large comparable corpora (English and Chinese Gigaword corpora) to gather more attributes for ENName and CHName. For example, since "Nick Grinde" is a "film director" while "Yi Wan . Si Te Lan Si Ji" is a "physicist" in these large contexts, we can filter out this incorrect name pair.

The detailed features converted from the above comments are summarized in Table II.

Table II
Validation Features for Name Translation

Comments	Features
1	co-occurrence, conditional probability and mutual information of *CHName* and *ENName* appearing in the same document from web-scale data
	conditional probability of *CHName* and *ENName* appearing in the same document from web-scale data
	mutual information of *CHName* and *ENName* appearing in the same document from web-scale data
2	Damerau-Levenshtein edit distance between the Pinyin form of *CHName* and *ENName*
3	overlap rate between the attributes of *CHName* and the attributes of *ENName* according to Wikipedia Infoboxes
	overlap rate between the attributes of *CHName* and the attributes of *ENName* according to IE results of large comparable corpora

C. Data and Scoring Metric

We used English and Chinese Wikipedias as of November 2010, including 10,355,225 English pages and 772,826 Chinese pages, and mined 5368 name pairs. A small set of 100 pairs is taken out as the development set for the human assessor to encode comments. The CGL assessor is then trained and tested on the remaining pairs by 5-folder cross-validation.

It is time consuming to evaluate the mined name pairs because sometimes the human assessor needs Web access to check the contexts of the pairs, especially when the translations are based on meanings instead of pronunciations. We implemented a baseline of mining name pairs from cross-lingual titles in Wikipedia as an incomplete answer key, and so we only need to ask two human assessors (not system developers) to do manual evaluation on our system generated pairs which are not in this answer key (1672 in total). A name pair is judged as correct if both of them are correctly extracted and one is the correct translation of the other. Such a semi-automatic method can speed up evaluation. On average each human assessor spent about 3 hours on evaluation.

D. Overall Performance

Table III shows Precision (P), Recall (R) and F-measure (F) scores before and after applying the CRL assessor on name translation pairs. As we can see from Table III, CGL achieved 28.7% absolute improvement on precision with a small loss (4.9%) in recall. In order to check how robust our approach is, we conducted the Wilcoxon Matched-Pairs Signed-Ranks Test on F-measures. The results show that we can reject the hypothesis that the improvements using CGL were random at a 99.8% confidence level.

Table III
Overall CGL Performance on Name Translation

Apply CGL	P (%)	R (%)	F (%)
before	69.3	1	81.9
after	98.0	95.1	**96.5**

V. Slot Filling

In this section, we shall apply CGL to a more challenging task of slot filling and investigate the detailed aspects of CGL by comparing it with other alternative methods.

A. Task Definition

In the slot filling task [15], attributes (or "slots") derived from Wikipedia infoboxes are used to create the initial (or reference) knowledge base (KB). A large collection of source news and web documents is then provided to the slot filling systems to expand the KB automatically.

The goal of slot filling is to collect information regarding certain attributes of a query from the corpus. The system must determine from this large corpus the values of specified attributes of the entity. Along with each slot answer, the system must provide the ID of a document which supports the correctness of this answer.

We choose three residence slots for person entities ("countries_of_residence", "stateorprovinces_of_residence" and "cities_of_residence") for our case study because they are one group of the most challenging slot types for which almost all systems perform poorly (less than 20% F-measure).

B. Baseline Systems

We use our slot filling system [7] which achieved highly competitive results (ranked at top 3 among 31 submissions from 15 teams) at the KBP2010 evaluation as our baseline. This system includes multiple pipelines in two categories: bottom-up IE based approaches (pattern matching and supervised classification) and a top-down Question Answering (QA) based approach that search for answers constructed from target entities and slot types. The overall system begins with an initial query processing stage where query expansion techniques are used to improve recall. The best answer candidate sets are generated from each of the individual pipelines and are combined in a statistical re-ranker. The resulting answer set, along with confidence values are then processed by a cross-slot reasoning step based on Markov Logic Networks [16], resulting in the final system outputs. In addition, the system also exploited external knowledge bases such as Freebase [17] and Wikipedia text mining for answer validation.

In order to check how robust the CGL assessor is, we also run it on some other anonymous systems in KBP2010 with representative performance (high, medium and low).

C. Comments and Feature Encoding

The detailed comments used for our slot filling experiment are as follows.

- Comment 1: “this answer is not a geo-political name”
  This comment is intended to address some obvious errors which could not be Geo-political (GPE) names in any contexts. In order to address this comment, we apply a very large gazetteer of GPE hierarchy (countries, states and cities) from the geonames website [2] for answer validation.
- Comment 2: “this answer is not supported by this document”
  Some answers obtained from Freebase may be incorrect because they are not supported by the source document. Answer validation was mostly conducted on the document basis, but for the residence slots we need to use sentence-level validation. In addition, some sentence segmentation errors occur in web documents. To address this comment, we apply a coreference resolution system [12] to the source document, and check whether any mention of the query entity and any mention of the candidate answer entity appear in the same sentence.
- Comment 3: “this answer is not a geo-political name in this sentence”
  Some ambiguous answers are not GPE names in certain contexts, such as “European Union”. To address this comment, we extract the context sentences including the query and answer mentions, and run a name tagger [18] to verify the candidate answer is a GPE name.
- Comment 4: “this answer conflicts with this system/other system’s output”
  When an answer from our system is not consistent with another answer which appears often in the pooled system responses, this comment suggests us to remove our answer. In order to address this comment, we implemented a feature based on hierarchical spatial reasoning. We conduct majority voting on all the available system responses, and collect the answers with global confidence values (voting weights) into a separate answer set ha. Then for any candidate answer a, we check the consistency between a and any member of ha by name coreference resolution and part-whole relation detection based on the gazetteer of GPE hierarchy as described in Comment 1. For example, if “U.S.” appears often in ha we can infer “Paris” is unlikely to be a correct answer for the same query; on the other hand if “New York” appears often in ha we can confirm “U.S.” as a correct answer.

The detailed features converted from the above comments are summarized in Table IV.

Table IV
VALIDATION FEATURES FOR SLOT FILLING

Comments	Features
1	whether the answer is in the geo-political gazetteer
2	whether any mention of the query entity and any mention of the answer entity appear in the same sentence using coreference resolution
3	whether the answer is a GPE name by running name tagging on the context sentence
4	whether the answer conflicts with the other answers which received high votes across systems using inferences through the GPE hierarchy

D. Data and Scoring Metric

During KBP2010, an initial answer key annotation was created by LDC through a manual search of the corpus, and then an independent adjudication pass was applied by LDC human assessors to assess these annotations together with pooled system responses to form the final gold-standard answer key. We incorporated the assessment comments for our system output on a separate development set (182 unique non-NIL answers in total) from KBP2010 training data set to train the automatic assessor. Then we conduct blind test on the KBP2010 evaluation data set which includes 1.7 million newswire and web documents. The final answer key for the blind test set includes 81 unique non-NIL answers for 49 queries.

The number of features we can exploit is limited by the unknown restrictions of individual systems. For example, some other systems used distant learning based answer validation and so could not provide specific context sentences. Since comment 2 and comment 3 require context sentences, we trained one assessor using all features and tested it on our own system. Then we trained another assessor using only comment 1 and 4 and tested it on three other systems representing different levels of performance.

Equivalent answers (such as "the United States" and "USA") are grouped into equivalence classes. Each system answer is rated as correct, wrong, or redundant (an answer which is equivalent to another answer for the same slot or an entry already in the knowledge base). Given these judgments, we calculate the precision, recall and F-measure of each system, as defined in [15].

E. Overall Performance

Table V shows the slot filling scores before and after applying the CGL assessors (because of the KBP Track requirements and policies, we could not mention the specific names of other systems). The Wilcoxon Matched-Pairs Signed-Ranks Test show we can reject the hypothesis that the improvements using CGL over our system were random at a 99.8% confidence level. It also indicates that the features encoded from comment 2 and comment 3, which require intermediate results such as context sentences helped boost the performance about 3.4%. We can see that although the other high-performing system may have used very different algorithms and resources from ours, our assessor still provided significant gains. Our approach improved the precision on each system (more than 200% relative gains) with some loss in recall. Since most comments focused on improving precision, F-measure gains for moderate-performing and low-performing systems were limited by their recall scores. This is similar to the human learning scenario where students from the same grade can learn more from each other than from different grades. In addition, the errors removed by our approach were distributed equally in newswire (48.9%) and web data (51.1%), which indicates the comments from human assessors reached a good degree of generalization across genres.

F. Cost and Contribution of Each Comment

The comments from the CGL assessor may reflect different aspects of the system. Therefore it will be interesting to investigate what types of comments are most useful and not costly. We did another experiment by applying one comment at a time into the assessor. Table VI shows the results along with the cost of generating and encoding each comment (i.e., knowledge transferring to its corresponding feature), which was carefully recorded by the human assessors.

Table V
OVERALL CGL PERFORMANCE ON SLOT FILLING

Slot Filling Systems		Apply CGL	P (%)	R (%)	F (%)
Our system		before	17.1	30.9	22.0
		after (f1+f4)	26.2	27.2	**26.7**
		after (full)	38.5	24.7	**30.1**
Other systems	High-Performing	before	13.7	29.6	18.8
		after (f1+f4)	40.9	22.2	**28.8**
	Moderate-Performing	before	12.2	7.4	9.2
		after (f1+f4)	35.7	6.2	**10.5**
	Low-Performing	before	6.7	3.7	4.8
		after (f1+f4)	50	3.7	**6.9**

Table VI
COST AND CONTRIBUTION OF EACH COMMENT

Comments		base-line	1	2	3	4
Perfor-mance	P (%)	17.1	17.6	26.4	26.7	25.6
	R (%)	30.9	30.9	28.4	28.4	27.2
	F (%)	22.0	22.4	27.4	27.5	26.3
Cost	#samples reviewed	-	3	3	3	3
	providing comments (minutes)	-	3	3	3	3
	encoding comments (minutes)	-	30	240	60	30

Table VI indicates that every feature made contributions to precision improvement. Comment 1 (gazetteer-based filtering) only provided limited gains mainly because our own system already extensively used similar gazetteers for answer filtering. This reflects a drawback of our comment generation procedure - the assessor had no prior knowledge about the approaches used in the systems. Comment 2 (using coreference resolution to check sentence occurrence) took most time to encode but also provides significant improvement. Comment 4 (consistency checking against responses with high votes) provided significant gains in precision (8.5%) but also some loss in recall (3.7%). The problem was that systems tend to make similar mistakes, and the human assessor was biased by those correct answers which appeared frequently in the pooled system output. However, Comment 4 was able to filter out many errors which are otherwise very difficult to detect. For example, because "Najaf" appears very often as a "cities_of_residence" in the pooled system responses, Comment 4 successfully removed six incorrect "countries_of_residence" answers for the same query: "Syrian", "Britain", "Iranian", "North Korea", "Saudi Arabia" and "United States". On the other hand, Comment 4 confirmed correct answers such as "New York" from "Brooklyn", "Texas" from "Dallas", "California" and "US" from "Los Angeles".

G. Impact of Data Size

We also did a series of runs to examine how our own system performed with different amounts of training data. These experiments are summarized in Figure 2. It clearly shows that the learning curve converges quickly. Therefore, we only need a very small amount of training data (36 samples, 20% of total) in order to obtain similar gains (6.8%) as using the whole training set.

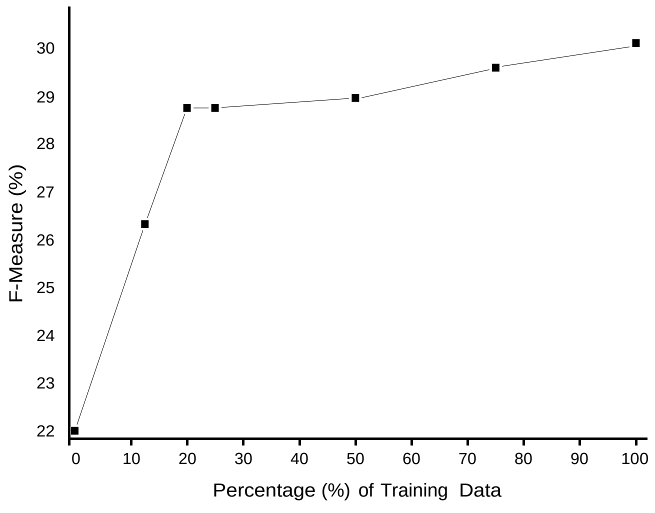

Figure 2. Impact of Training Data Size

H. Speed up Human Assessment

Human assessment for slot filling is also a costly task because it requires the annotators to judge each answer against the associated source document. Since our CGL approach achieved positive impact on system output, can it be used to as feedback to speed up human assessment? We applied the CGL assessor trained from comment 1 and comment 4 to the top 13 KBP systems for KBP2010 evaluation set. We automatically ranked the pooled system responses of residence slots according to their confidence values from high to low.

For comparison, we also exploited the following methods:

- Baseline

  As a baseline, we ranked the responses according to the alphabetical order of slot type, query ID, query name and answer string and doc ID. This is the same approach used by LDC human annotators for assessing KBP2010 system responses.

- Oracle (Upper-Bound)

  We used an oracle (for upper-bound analysis) by always assessing all correct answers first.

Figure 3 summarizes the results from the above 4 approaches. For this figure, we assume a labor cost for assessment proportional to the number of non-NIL items assessed. Note that all redundant answers are also included in these counts because human assessors also spent time on assessing them. This is only approximately correct; it may be faster (per response) to assess more responses to the same slot. The common end point of curves represents the cost and benefit of assessing all system responses. We can see that if we employ the CGL assessor and apply some cut-off, the process can be dramatically more efficient than the regular baseline based on alphabetical order. For example, in order to get 79 correct answers (76% of total), CGL approach took human assessors only 5.5 hours, while the baseline approach took 13.4 hours.

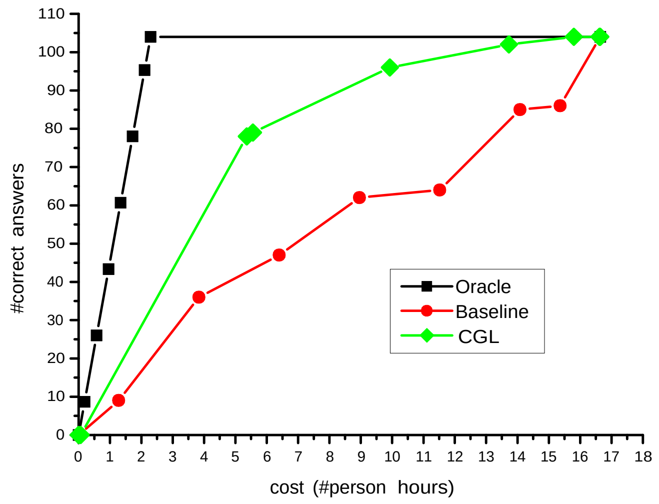

Figure 3. Human Assessment Method Comparison

I. Discussion

An alternative approach to validate answers is to use textual entailment techniques as in the RTE-KBP2010 validation pilot study [4], which was partly inspired by CLEF Question Answering task [19]. This task consists of determining whether a candidate answer (hypothesis "H") is supported in the associated source document (text "T") using entailment techniques. For the residence slots, we are considering in this paper, they treat each context document as a "T", and apply pre-defined sentence templates such as "[Query] lived in [Answer]" to compose a "H" from system output. Entailment and reasoning methods from the TAC-RTE2010 systems are then applied to validate whether "H" is true or false according to "T". These RTE-KBP systems are limited to individual H-T instances and optimized only on a subset of the pooled system responses. As a result, they aggressively filtered many correct answers and did not provide improvement on most slot filling systems (including the representative ones we used for our experiment). In contrast, our CGL approach has the advantage of exploiting the generalized knowledge and feedback from assessors across all queries and systems.

VI. Conclusion and Future Work

To sum up, we have described a new validation approach called comment-guided learning (CGL). We demonstrated the power and generality of this approach on two very different applications: name translation and slot filling. Our approach significantly improved the performance of system responses and speeded up human assessment. It also outperformed some traditional validation methods, which, unlike ours, involved a great deal of feature engineering effort. The novelty of our approach lies in its declarative use of assessment feedback which may address some typical errors that a system tends to make. Some of such feedback will be otherwise difficult to acquire for feature encoding (e.g., Comment 3 in name translation and Comment 4 in slot filling). On the other hand, the simplicity of our approach lies in its low cost because it incorporates the bi-product of human assessment, namely their comments and explanations, instead of tedious instance-based human correction into the learning process. In this way the human assessor's knowledge is naturally transferred to the automatic assessor. Hence, CGL is amenable to implement but pertinent to a series of common errors identified.

In the future, we are interested in extending this idea to improve other NLP applications and integrating it with human reasoning. The current setup mainly improved precision but we also plan to embrace the idea of revertible query in question answering literature (e.g., [20]) and relation graph traverse to enhance recall. Ultimately we intend to investigate automatic ways to prioritize comments and convert comments to features so that we can better simulate the role of teacher in human learning.

Acknowledgments

This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), the U.S. NSF CAREER Award under Grant IIS-0953149, the U.S. NSF EAGER grant No. 1144111 and PSC-CUNY Research Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] V. Vapnik, "Learning with Teacher: Learning Using Hidden Information," in Proc. International Joint Conference on Neural Networks, 2009.

[2] R. Castro, C. Kalish, R. Nowak, R. Qian, T. Rogers, and X. Zhu "Human Active Learning," in Proc. NIPS2008, 2008.

[3] E. Bril, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging," in Computational Linguistics, vol. 21, no. 1, 1995.

[4] K. Williams, C. Dozier, and A. McCulloh, "Learning transformation rules for semantic role labeling," in Proc. CoNLL-2004, 2004, pp. 134-137.

[5] Y. Al-Onaizan and K. Knight, "Translating named entities using monolingual and bilingual resources," in Proc. ACL2002, 2002, pp. 400-408.

[6] L. Bentivogli, P. Clark, I. Dagan, H. Dang, and D. Giampiccolo, "The sixth pascal recognizing textual entailment challenge," in Proc. TAC 2010 Workshop, 2010.

[7] Z. Chen, S. Tamang, A. Lee, X. Li, W.-P. Lin, M. Snover, J. Artiles, M. Passantino, and H. Ji, "Cuny-blender tac-kbp2010 entity linking and slot filling system description," in Proc. TAC 2010 Workshop, 2010.

[8] V. Castelli, R. Florian, and D. Han, "Slot filling through statistical processing and inference rules," in Proc. TAC 2010 Workshop, 2010.

[9] G. Bouma, S. Duarte, and Z. Islam, "Cross-lingual Alignment and Completion of Wikipedia Templates," in Proc. the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies, 2009, pp. 21-29.

[10] D. Lin, S. Zhao, B. V. Durme, and M. Pasca, "Mining Parenthetical Translations from the Web by Word Alignment," in Proc. ACL2008, 2008, pp. 994-1002.

[11] G. You, S. Hwang, Y. Song, L. Jiang, and Z. Nie, "Mining Name Translations from Entity Graph Mapping," in Proc EMNLP2010, 2010, pp. 430-439.

[12] W.-P. Lin, M. Snover, and H. Ji, "Unsupervised language-independent name translation mining from wikipedia infoboxes," in Proc. EMNLP2011 Workshop on Unsupervised Learning for NLP, 2011, pp. 43-52.

[13] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in Soviet Physics Doklady, 1966.

[14] H. Ji, D. Westbrook, and R. Grishman, "Using semantic relations to refine coreference decisions," in Proc. HLT/EMNLP 05, 2005, pp. 17-24.

[15] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis, "Overview of the tac 2010 knowledge base population track," in Proc. TAC 2010 Workshop, 2010.

[16] M. Richardson and P. Domingos, "Markov logic networks," in Machine Learning, 2006.

[17] K. Bollacker, R. Cook, and P. Tufts, "Freebase: A shared database of structured general human knowledge," in Proc. National Conference on Artificial Intelligence (Volume 2), 2007.

[18] R. Grishman, D. Westbrook, and A. Meyers, "Nyu's english ace 2005 system description," in Proc. ACE2005, 2005.

[19] A. Penas, A. Rodrigo, V. Sama, and F. Verdejo, "Testing the reasoning for question answering validation," in Journal of Logic and Computation, 2007.

[20] J. Prager, P. Duboue, and J. Chu-Carrol, "Improving qa accuracy by question inversion," in Proc. ACL-COLING2006, 2006, pp. 1073-1080.