Nuggeteer: Automatic Nugget-Based Evaluation using Descriptions and Judgements

Gregory Marton **Alexey Radul**
Infolab Group, MIT CSAIL
Cambridge, MA 02139
{gremio,axch}@mit.edu

Abstract

The TREC Definition and Relationship questions are evaluated on the basis of information nuggets that may be contained in system responses. Human evaluators provide informal descriptions of each nugget, and judgements (assignments of nuggets to responses) for each response submitted by participants. While human evaluation is the most accurate way to compare systems, approximate automatic evaluation becomes critical during system development.

We present Nuggeteer, a new automatic evaluation tool for nugget-based tasks. Like the first such tool, Pourpre, Nuggeteer uses words in common between candidate answer and answer key to approximate human judgements. Unlike Pourpre, but like human assessors, Nuggeteer creates a judgement for each candidate-nugget pair, and can use existing judgements instead of guessing. This creates a more readily interpretable aggregate score, and allows developers to track individual nuggets through the variants of their system. Nuggeteer is quantitatively comparable in performance to Pourpre, and provides qualitatively better feedback to developers.

1 Introduction

The TREC Definition and Relationship questions are evaluated on the basis of information nuggets, abstract pieces of knowledge that, taken together, comprise an answer. Nuggets are described informally, with abbreviations, misspellings, etc., and each is associated with an importance judgement: 'vital' or 'okay'.[1] In some sense, nuggets are like WordNet synsets, and their descriptions are like glosses. Responses may contain more than one nugget—when they contain more than one piece of knowledge from the answer. The median scores of today's systems are frequently zero; most responses contain no nuggets (Voorhees, 2005).

Human assessors decide what nuggets make up an answer based on some initial research and on pools of top system responses for each question. Answer keys list, for each nugget, its id, importance, and description; two example answer keys are shown in Figures 1 and 2. Assessors make binary decisions about each response, whether it contains each nugget. When multiple responses contain a nugget, the assessor gives credit only to the (subjectively) best response.

Using the judgements of the assessors, the final score combines the recall of the available vital nuggets, and the length (discounting whitespace) of the system response as a proxy for precision. Nuggets valued 'okay' contribute to precision by increasing the length allowance, but do not contribute to recall. The scoring formula is shown in Figure 3.

[1] Nuggeteer implements the *pyramid* scoring system from (Lin and Demner-Fushman, 2006), designed to soften the dis-

Qid 87.8: 'other' question for target Enrico Fermi

1	*vital*	belived in partical's existence and named it neutrino
2	*vital*	Called the atomic Bomb an evil thing
3	*okay*	Achieved the first controlled nuclear chain reaction
4	*vital*	Designed and built the first nuclear reactor
5	*okay*	Concluded that the atmosphere was in no real danger before Trinity test
6	*okay*	co-developer of the atomic bomb
7	*okay*	pointed out that the galaxy is 100,000 light years across

Figure 1: The "answer key" to an "other" question from 2005.

The analyst is looking for links between Colombian businessmen and paramilitary forces. Specifically, the analyst would like to know of evidence that business interests in Colombia are still funding the AUC paramilitary organization.

1	*vital*	Commander of the national paramilitary umbrella organization claimed his group enjoys growing support from local and international businesses
2	*vital*	Columbia's Chief prosecutor said he had a list of businessmen who supported right-wing paramilitary squads and warned that financing outlawed groups is a criminal offense
3	*okay*	some landowners support AUC for protections services
4	*vital*	Rightist militias waging a dirty war against suspected leftists in Colombia enjoy growing support from private businessmen
5	*okay*	The AUC makes money by taxing Colombia's drug trade
6	*okay*	The ACU is estimated to have 6000 combatants and has links to government security forces.
7	*okay*	Many ACU fighters are former government soldiers

Figure 2: The "answer key" to a relationship question.

Let

r	# of *vital* nuggets returned in a response
a	# of *okay* nuggets returned in a response
R	# of *vital* nuggets in the answer key
l	# of non-whitespace characters in the entire answer string

Then

$$\text{"recall"}\ \mathcal{R} = r/R$$

$$\text{"allowance"}\ \alpha = 100 \times (r + a)$$

$$\text{"precision"}\ \mathcal{P} = \begin{cases} 1 & \text{if } l < \alpha \\ 1 - \frac{l-\alpha}{l} & \text{otherwise} \end{cases}$$

Finally, the $F(\beta) = \dfrac{(\beta^2 + 1) \times \mathcal{P} \times \mathcal{R}}{\beta^2 \times \mathcal{P} + \mathcal{R}}$

Figure 3: Official definition of F-measure.

Automatic evaluation of systems is highly desirable. Developers need to know whether one system performs better or worse than another. Ideally, they would like to know which nuggets were lost or gained. Because there is no exhaustive list of snippets from the document collection that contain each nugget, an exact automatic solution is out of reach. Manual evaluation of system responses is too time consuming to be effective for a development cycle.

The Qaviar system first described an approximate automatic evaluation technique using keywords, and Pourpre was the first publicly available implementation for these nugget-based tasks. (Breck et al., 2000; Lin and Demner-Fushman, 2005). Pourpre calculates an *idf*- or count-based, stemmed, unigram similarity between each nugget description and each

tinction between 'vital' and 'okay'.

candidate system response. If this similarity passes a threshold, then it uses this similarity to assign a partial value for recall and a partial length allowance, reflecting the uncertainty of the automatic judgement. Importantly, it yields a ranking of systems very similar to the official ranking (See Table 2).

Nuggeteer offers three important improvements:

- interpretability of the scores, as compared to official scores,
- use of known judgements for exact information about some responses, and
- information about individual nuggets, for detailed error analysis.

Nuggeteer makes scores interpretable by making binary decisions about each nugget and each system response, just as assessors do, and then calculating the final score in the usual way. We will show that Nuggeteer's absolute error is comparable to human error, and that the 95% confidence intervals Nuggeteer reports are correct around 95% of the time.

Nuggeteer assumes that if a system response was ever judged by a human assessor to contain a particular nugget, then other identical responses also contain that nugget. When this is not true among the human judgements, we claim it is due to annotator error. This assumption allows developers to add their own judgements and have the responses they've adjudicated scored "exactly" by Nuggeteer.

These features empower developers to track not only the numeric value of a change to their system, but also its effect on retrieval of each nugget.

2 Approach

Nuggeteer builds one binary classifier per nugget for each question, based on n-grams (up to trigrams) in the description and optionally in any provided judgement files. The classifiers use a weight for each n-gram, an informativeness measure for each n-gram, and a threshold for accepting a response as bearing the nugget.

2.1 N-gram weight

The *idf*-based weight for an n-gram $w_1...w_n$ is the sum of unigram *idf* counts from the AQUAINT corpus of English newspaper text, the corpus from which responses for the TREC tasks are drawn. We did not explore using n-gram *idf*s. A tf component is not meaningful because the data are so sparse.

2.2 Informativeness

Let G be the set of nuggets for some question. Informativeness of an n-gram for a nugget g is calculated based on how many other nuggets in that question ($\in G$) contain the n-gram. Let

$$i(g, w_1...w_n) = \begin{cases} 1 & \text{if } count(g, w_1..w_n) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $count(g, w_1...w_n)$ is the number of occurrences of the n-gram in responses containing the nugget g.

Then informativeness is:

$$I(g, w_1...w_n) = 1 - \frac{\sum_{g' \in G} i(g', w_1...w_n)}{|G|} \quad (2)$$

This captures the Bayesian intuition that the more outcomes a piece of evidence is associated with, the less confidence we can have in predicting the outcome based on that evidence.

2.3 Judgement

Nuggeteer does not guess on responses which have been judged by a human to contain a nugget, or those which have unambiguously judged not to, but assigns the known judgement.[2]

For unseen responses, we determine the n-gram recall for each nugget g and candidate response $w_1...w_l$ by breaking the candidate into n-grams and finding the sum of scores:

$$Recall(g, w_1...w_l) = \sum_{k=0}^{n-1} \sum_{i=0}^{l-k} W(g, w_i...w_{i+k}) * I(g, w_i...w_{i+k}) \quad (3)$$

The candidate is considered to contain all nuggets whose recall exceeds some threshold. Put another

[2] If a response was submitted, and no response from the same system was judged to contain a nugget, then the response is considered to not contain the nugget. We normalized whitespace and case for matching previously seen responses.

way, we build an n-gram language model for each nugget, and assign those nuggets whose predicted likelihood exceeds a threshold.

When several responses contain a nugget, Nuggeteer picks the *first* (instead of the best, as assessors can) for purposes of scoring.

2.4 Parameter Estimation

We explored a number of parameters in the scoring function: stemming, n-gram size, *idf* weights vs. count weights, and the effect of removing stopwords. We tested all 24 combinations, and for each experiment, we cross-validated by leaving out one submitted system, or where possible, one submitting institution (to avoid training and testing on potentially very similar systems).[3]

Each experiment was performed using a range of thresholds for Equation 3 above, and we selected the best performing threshold for each data set.[4] Because the threshold was selected after cross-validation, it is exposed to overtraining. We used a single global threshold to minimize this risk, but we have no reason to think that the thresholds for different nuggets are related.

Selecting thresholds as part of the training process can maximize accuracy while eliminating overtraining. We therefore explored Bayesian models for automatic threshold selection. We model assignment of nuggets to responses as caused by the scores according to a noisy threshold function, with separate false positive and false negative error rates. We varied thresholds and error rates by entire dataset, by question, or by individual nugget, evaluating them using Bayesian model selection.

3 The Data

For our experiments, we used the definition questions from TREC2003, the 'other' questions from TREC2004 and TREC2005, and the relationship questions from TREC2005. (Voorhees, 2003; Voorhees, 2004; Voorhees, 2005) The distribution of nuggets and questions is shown for each data set in Table 1. The number of nuggets by number of system responses assigned that nugget (difficulty of nuggets, in a sense) is shown in Figure 4. More than a quarter of relationship nuggets were not found by any system. Among all data sets, many nuggets were found in none or just a few responses.

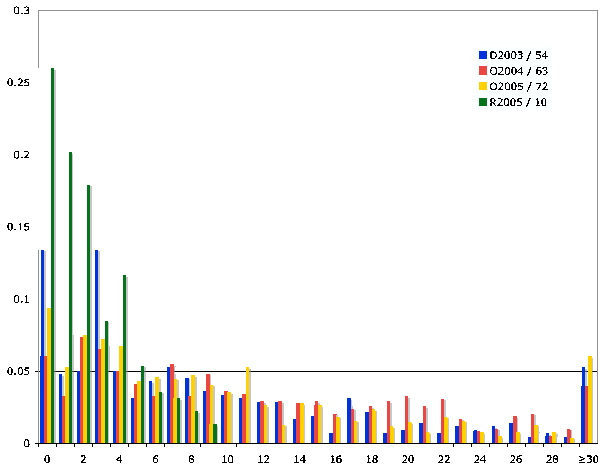

Figure 4: Percents of nuggets, binned by the number of systems that found each nugget.

4 Results

We report correlation (R^2), and Kendall's τ_b, following Lin and Demner-Fushman. Nuggeteer's scores are in the same range as real system scores, so we also report average root mean squared error from the official results. We 'corrected' the official judgements by assigning a nugget to a response if that response was judged to contain that nugget in any assessment for any system.

4.1 Comparison with Pourpre

(Lin et al., 2005) report Pourpre and Rouge performance with Pourpre optimal thresholds for TREC definition questions, as reproduced in Table 2. Nuggeteer's results are shown in the last column.[5]

Table 3 shows a comparison of Pourpre and Nuggeteer's correlations with official scores. As ex-

[3]For TREC2003 and TREC2004, the run-tags indicate the submitting institution. For TREC2005 we did not run the non-anonymized data in time for this submission. In the TREC2005 Relationship task, RUN-1 was withdrawn.

[4]Thresholds for Pourpre were also selected this way.

[5]We report only micro-averaged results, because we wish to emphasize the interpretability of Nuggeteer scores. While the correlations of macro-averaged scores with official scores may be higher (as seems to be the case for Pourpre), the actual values of the micro-averaged scores are more interpretable because they include a variance.

	#ques	#vital	#okay	#n/q	#sys	#r/s	#r/q/s
D 2003:	50	207	210	9.3± 1.0	54	526± 180	10.5± 1.2
O 2004:	64	234	346	10.1± .7	63	870± 335	13.6± 0.9
O 2005:	75	308	450	11.1± .6	72	1277± 260[a]	17.0± 0.6[a]
R 2005:	25	87	136	9.9± 1.6	10	379± 222[b]	15.2± 1.6[b]
[a] excluding RUN-135: 410,080 responses							5468 ± 5320
[b] excluding RUN-7: 6436 responses							257 ± 135

Table 1: For each data set (D="definition", O="other", R="relationship"), the number of questions, the numbers of vital and okay nuggets, the average total number of nuggets per question, the number of participating systems, the average number of responses per system, and the average number of responses per question over all systems.

Run	POURPRE				ROUGE		NUGGETEER
	micro, cnt	macro, cnt	micro, *idf*	macro, *idf*	default	stop	nostem, bigram, micro, *idf*
D 2003 ($\beta = 3$)	0.846	**0.886**	0.848	0.876	0.780	**0.816**	**0.879**
D 2003 ($\beta = 5$)	0.890	**0.878**	0.859	0.875	0.807	**0.843**	**0.849**
O 2004 ($\beta = 3$)	0.785	**0.833**	0.806	0.812	0.780	**0.786**	**0.898**
O 2005 ($\beta = 3$)	0.598	**0.709**	0.679	0.698	0.662	**0.670**	**0.858**
R 2005 ($\beta = 3$)		**0.697**					**1**

Table 2: Kendall's τ correlation between rankings generated by POURPRE/ROUGE/NUGGETEER and official scores, for each data set (D="definition", O="other", R="relationship"). τ=1 means same order, τ=-1 means reverse order. Pourpre and Rouge scores reproduced from (Lin and Demner-Fushman, 2005).

Run	POURPRE	NUGGETEER	
	R^2	R^2	$\sqrt{mse}$
D 2003 ($\beta = 3$)	0.963	0.966	0.067
D 2003 ($\beta = 5$)	0.965	0.971	0.077
O 2004 ($\beta = 3$)	0.929	0.982	0.026
O 2005 ($\beta = 3$)	0.916	0.952	0.026
R 2005 ($\beta = 3$)	0.764	0.993	0.009

Table 3: Correlation (R^2) and Root Mean Squared Error ($\sqrt{mse}$) between scores generated by Pourpre/Nuggeteer and official scores, for the same settings as the τ comparison above.

pected from the Kendall's τ comparisons, Pourpre's correlation is about the same or higher in 2003, but fares progressively worse in the subsequent tasks.

To ensure that Pourpre scores correlated sufficiently with official scores, Lin and Demner-Fushman used the difference in official score between runs whose ranks Pourpre had swapped, and showed that the majority of swaps were between runs whose official scores were less than the 0.1 apart, a threshold for assessor agreement reported in (Voorhees, 2003).

Nuggeteer scores are not only correlated with, but actually meant to approximate, the assessment scores; thus we can use a stronger evaluation: root mean squared error of Nuggeteer scores against official scores. This estimates the average difference between the Nuggeteer score and the official score, and at 0.077, the estimate is below the 0.1 threshold. This evaluation is meant to show that the scores are "good enough" for experimental evaluation, and in Section 4.4 we will substantiate Lin and Demner-Fushman's observation that higher correlation scores may reflect overtraining rather than actual improvement.

Accordingly, rather than reporting the best Nuggeteer scores (Kendall's τ and R^2) above, we follow Pourpre's lead in reporting a single variant (no stemming, bigrams) that performs well across the data sets. As with Pourpre's evaluation, the par-

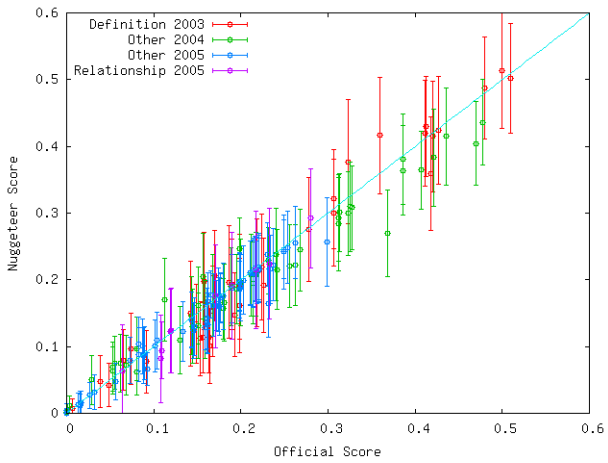

Figure 5: Scatter graph of official scores plotted against Nuggeteer scores (*idf* term weighting, no stemming, bigrams) for each data set (all F-measures have $\beta = 3$), with the Nuggeteer 95% confidence intervals on the score. Across the four datasets, 6 systems (3%) have an official score outside Nuggeteer's 95% confidence interval.

ticular thresholds for each year are experimentally optimized. A scatter plot of Nuggeteer performance on the definition tasks is shown in Figure 5.

4.2 N-gram size and stemming

A hypothesis advanced with Pourpre is that bigrams, trigrams, and longer n-grams will primarily account for the fluency of an answer, rather than its semantic content, and thus not aid the scoring process. We included the option to use longer n-grams within Nuggeteer, and have found that using bigrams can yield very slightly better results than using unigrams. From inspection, bigrams sometimes capture named entity and grammatical order features.

Experiments with Pourpre showed that stemming hurt slightly at peak performances. Nuggeteer has the same tendency at all n-gram sizes.

Figure 6 compares Kendall's τ over the possible thresholds, n-gram lengths, and stemming. The choice of threshold matters by far the most.

4.3 Term weighting and stopwords

Removing stopwords or giving unit weight to all terms rather than an *idf*-based weight made no substantial difference in Nuggeteer's performance.

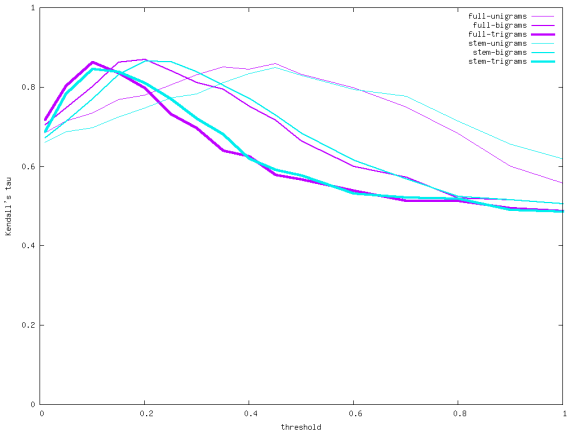

Figure 6: Fixed thresholds vs. Kendall's τ for unigrams, bigrams, or trigrams averaged over the three years of definition data using $F(\beta = 3)$.

Model	$\log_{10} P(\text{Data}\|\text{Model})$
optimally biased coin	-2780
global threshold	-2239
per-question thresholds	-1977
per-nugget thresholds	**-1546**
per-nugget errors and thr.	-1595

Table 4: The probabilities of the data given several models: a baseline coin, three models of different granularity with globally specified false positive and negative error rates, and a model with too many parameters, where even the error rates have per-nugget granularity. We select the most probable model, the per-nugget threshold model.

4.4 Thresholds

We experimented with Bayesian models for automatic threshold selection. In the models, a system response contains or does not contain each nugget as a function of the response's Nuggeteer score plus noise. Table 4 shows that, as expected, the best models do not make assumptions about thresholds being equal within a question or dataset. It is interesting to note that Bayesian inference catches the overparametrization of the model where error rates vary per-nugget as well. In essence, we do not need those additional parameters to explain the variation in the data.

The τ of the best selection of parameters on the 2003 data set using the model with one threshold per

nugget and global errors is 0.837 ($\sqrt{mse}$=0.037). We have indeed overtrained the best threshold for this dataset (compare τ=0.879, $\sqrt{mse}$=0.067 in Tables 2 and 3), suggesting that the numeric differences in Kendall's Tau shown between the Nuggeteer, Pourpre, and Rouge systems are not indicative of true performance. The Bayesian model promises settings free of overtraining, and thus more accurate judgements in terms of $\sqrt{mse}$ and individual nugget classification accuracy.

4.5 Training on System Responses

Intuitively, if a fact is expressed by a system response, then another response with similar n-grams may also contain the same fact. To test this intuition, we tried expanding our judgement method (Equation 3) to select the maximum judgement score from among those of the nugget description and each of the system responses judged to contain that nugget.

Unfortunately, the assessors did not mark which *portion* of a response expresses a nugget, so we also find spurious similarity, as shown in Figure 7. The final results are not conclusively better or worse overall, and the process is far more expensive.

We are currently exploring the same extension for multiple "nugget descriptions" generated by manually selecting the appropriate portions of system responses containing each nugget.

4.6 Judgment Precision and Recall

Because Nuggeteer makes a nugget classification for each system response, we can report precision and recall on the nugget assignments. Table 5 shows Nuggeteer's agreement rate with assessors on whether each response contains a nugget. [6]

4.7 Novel Judgements

Approximate evaluation will tend to undervalue new results, simply because they may not have keyword overlap with existing nugget descriptions. We are therefore creating tools to help developers manually assess their system outputs.

As a proof of concept, we ran Nuggeteer on the best 2005 "other" system (not giving Nuggeteer the official judgements), and manualy corrected its guesses.[7] Assessment took about 6 hours, and our judgements had precision of 78% and recall of 90%, for F-measure 0.803± 0.065 (compare Table 5). The official score of .299 was still within the confidence interval, but now on the high side rather than the low (.257± .07), because we found the answers quite good. In fact, we were often tempted to add new nuggets! We later learned that it was a manual run, produced by a student at the University of Maryland.

Data set	**best** F($\beta = 1$)	**default** F($\beta = 1$)
2003 defn	0.68± .01	0.66± .02
2004 other	0.73± .01	0.70± .01
2005 other	0.87± .01	0.86± .01
2005 reln	0.75± .04	0.72± .05

Table 5: Nuggeteer agreement with official judgements, under best settings for each year, and under the default settings.

5 Discussion

Pourpre pioneered automatic nugget-based assessment for definition questions, and thus enabled a rapid experimental cycle of system development. Nuggeteer improves on that functionality, and critically adds:

- an interpretable score, comparable to official scores, with near-human error rates,
- a reliable confidence interval on the estimated score,
- scoring known responses exactly,
- support for improving the accuracy of the score through additional annotation, and
- a more robust training process

We have shown that Nuggeteer evaluates the definition and relationship tasks with comparable rank swap rates to Pourpre. We explored the effects of stemming, term weighting, n-gram size, stopword removal, and use of system responses for training, all with little effect. We showed that previous methods of selecting a threshold overtrained, and have

[6] Unlike human assessors, Nuggeteer is not able to pick the "*best*" response containing a nugget if multiple responses have it, and will instead pick the *first*, so these values are artifactually low. However, 2005 results may be high because these results reflect anonymized runs.

[7] We used a low threshold to make the task mostly correcting and less searching. This is clearly not how assessors should work, but is expedient for developers.

question id 1901, *response rank* 2, *response score* 0.14

response text: `best american classical music bears its stamp: witness aaron copland, whose "american-sounding" music was composed by a` (the response was a sentence fragment)

assigned nugget description: born brooklyn ny 1900

bigram matches: "american classical", "american-sounding music", "best american", "whose american-sounding", "witness aaron", "copland whose", "stamp witness", ...

response containing the nugget: `Even the best American classical music bears its stamp: witness Aaron Copland, whose ``American-sounding'' music was composed by a Brooklyn-born Jew of Russian lineage who studied in France and salted his scores with jazz-derived syncopations, Mexican folk tunes and cowboy ballads.` NYT19981210.0106

Figure 7: This answer to the definition question on Aaron Copeland is assigned the nugget "born brooklyn ny 1900" at a recall score well above that of the background, despite containing none of those words.

briefly described a promising way to select finer-grained thresholds automatically.

Our experiences in using judgements of system responses point to the need for a better annotation of nugget content. It is possible to give Nuggeteer multiple nugget descriptions for each nugget. Manually extracting the relevant portions of correctly-judged system responses may not be an overly arduous task, and may offer higher accuracy. It would be ideal if the community—including the assessors—were able to create and promulgate a gold-standard set of nugget descriptions for previous years.

Nuggeteer currently supports evaluation for the TREC definition, 'other', and relationship tasks, for the AQUAINT opinion pilot [8], and is under development for the DARPA GALE task [9].

6 Acknowledgements

We would like to thank Jimmy Lin and Dina Demner-Fushman for valuable discussions, for Figure 3, and Table 2, and for creating Pourpre. Thanks to Ozlem Uzuner and Sue Felshin for valuable comments on earlier drafts of this paper and to Boris Katz for his inspiration and support.

[8] http://www-24.nist.gov/projects/aquaint/opinion.html

[9] http://www.darpa.mil/ipto/programs/gale

References

Eric J. Breck, John D. Burger, Lisa Ferro, Lynette Hirschman, David House, Marc Light, and Inderjeet Mani. 2000. How to evaluate your question answering system every day ... and still get real work done. In *Proceedings of the second international conference on Language Res ources and Evaluation (LREC2000)*.

Jimmy Lin and Dina Demner-Fushman. 2005. Automatically evaluating answers to definition questions. In *Proceedings of HLT-EMNLP*.

Jimmy Lin and Dina Demner-Fushman. 2006. Will pyramids built of nuggets topple over? In *Proceedings of HLT-NAACL*.

Jimmy Lin, Eileen Abels, Dina Demner-Fushman, Douglas W. Oard, Philip Wu, and Yejun Wu. 2005. A menagerie of tracks at maryland: HARD, Enterprise, QA, and Genomics, oh my! In *Proceedings of TREC*.

Ellen Voorhees. 2003. Overview of the TREC 2003 question answering track.

Ellen Voorhees. 2004. Overview of the TREC 2004 question answering track.

Ellen Voorhees. 2005. Overview of the TREC 2005 question answering track.