Nuggeteer: Automatic Nugget-Based Evaluation using Descriptions and Judgements

Gregory Marton

February 22, 2006

Abstract

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.

The current best automatic evaluation for these kinds of questions is Pourpre. Pourpre uses a stemmed unigram similarity of responses with nugget descriptions, yielding an aggregate result that is difficult to interpret, but is useful for relative comparison. Nuggeteer, by contrast, uses both the human descriptions and the human judgements, and makes binary decisions about each response, so that the end result is as interpretable as the official score.

I explore n-gram length, use of judgements, stemming, and term weighting, and provide a new algorithm quantitatively comparable to, and qualitatively better than, the state of the art.

1 Introduction

TREC Definition and Relationship questions are evaluated on the basis of information nuggets that collectively form the sought-after information for a question. Nuggets are pieces of knowledge, represented by an id and an informal description (a note-to-self, 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 [7].

Human assessors decide what nuggets make up an answer based on pools of top system responses for each question, and on some initial research. The answer key for a question lists each nugget id, nugget importance, and nugget 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.

[1] The distinction between vital and okay has recently come under question; see [4].

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.

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.[2]

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 [1] for factoid evaluation is very similar in spirit and implementation to Nuggeteer. Factoid questions today require short exact answers, e.g. "John Wilkes Booth" for "Who shot Lincoln?", and so are better suited for evaluation using answer patterns. But factoid questions in TREC-8 were similar to definition questions in scoring longer (250 byte) answer strings, rather than exact answers. Qaviar first introduced evaluation using unigram recall, and reported close-to-human performance in both individual judgements and overall system rankings. We have not yet directly compared Qaviar's performance with Nuggeteer's, but we expect that adapting Qaviar to the definition and relationship task would yield comparable results. Nuggeteer has facilities for using n-grams of length greater than one, and this may yield a slight advantage.

Pourpre was the first, and until now the only system that provided approximate automatic evaluation of system responses [3]. Pourpre calculates an *idf*- or count-based, stemmed, unigram similarity between each nugget description and each 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 on page 7).

[2]Thanks to Jimmy Lin and Dina Demner-Fushman for Figure 3, and Table 2.

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.

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. I show that the root mean squared error of Nuggeteer scores with official scores is small, and that most official scores are within the 95% confidence interval that Nuggeteer reports (see Section 4.5).

Nuggeteer makes the assumption 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. This is not always true among the human evaluations, but we claim that those cases are due to annotator error. Using this assumption, Nuggeteer can "exactly" score already-seen responses in new system outputs.

Nuggeteer allows developers to test for the presence of individual nuggets by providing its guesses in new "assessment" files. If they adjudicate these responses, they can add them to the "known" set, improving the accuracy of their Nuggeteer scores.

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

$$\begin{aligned} \text{recall } (\mathcal{R}) &= r/R \\ \text{allowance } (\alpha) &= 100 \times (r + a) \\ \text{precision } (\mathcal{P}) &= \begin{cases} 1 & \text{if } l < \alpha \\ 1 - \frac{l-\alpha}{l} & \text{otherwise} \end{cases} \end{aligned}$$

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

$\beta = 5$ in TREC 2003, $\beta = 3$ in TREC 2004.

Figure 3: Official definition of F-measure.

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 $w_1 ... w_n$, in a response containing nugget g, uses *idf* counts from the AQUAINT corpus of English newspaper text, the corpus from which responses for the particular TREC tasks of interest are drawn.[3]

$$W(g, w_1 ... w_n) = \sum_{1}^{n} idf(w_i) \tag{1}$$

[3]Adding a tf component is not meaningful because the data are so sparse.

2.2 Informativeness

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

$$i(g, w_1...w_n) = \begin{cases} 1 & \text{if } count(g, w_1..w_n) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

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:

$$I(g, w_1...w_n) = 1 - \frac{\sum_{g' \in G} i(g', w_1...w_n)}{|G|} \quad (3)$$

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

I 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:

$$Recall(g, w_1...w_l) = \quad (4)$$
$$\sum_{k=0}^{n-1} \sum_{i=-k}^{l} W(g, w_i...w_{i+k}) * I(g, w_i...w_{i+k})$$

I rank the nuggets by recall and assign all nuggets whose recall is above a global threshold.[4]

Essentially, 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 arbitrarily picks the *first* (instead of the best, as assessors can) for purposes of scoring.

2.4 Multiple Descriptions

When learning n-grams from judgements, we treat each assigned system response as an additional nugget description. In this case informativeness is based on the union of all description words for each nugget. Recall, on the other hand, is the highest recall of any individual description.

We tried but rejected another approach which would have given a recall score based on the "product of doubts": $1 - \sum_{d \in D} 1 - \text{recall}_d$ where D is the set of descriptions. It is usually incorrect to treat recall from one description as interchangeable with recall from another.

[4]When learning n-grams from judgements, recall must also exceed that of a special *null* nugget, representing a background language model for the question, built from unassigned responses.

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. [5, 6, 7] 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.

	#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	$10.1 \pm .7$	63	870 ± 335	13.6 ± 0.9
O 2005:	75	308	450	$11.1 \pm .6$	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 containing: 410,080 responses							5468 ± 5320
[b] excluding RUN-7 containing: 6436 responses							257 ± 135

Table 1: For each data set, 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. I present RUN-135 and RUN-7 separately, only in this table, because of their extraordinary length.

4 Experiments

To test the performance of each method, I cross-validated over system outputs. For TREC2003 and TREC2004, the run-tags give an indication of the institution that submitted each run. Runs from the same institution may be similar, so when testing each run, I trained only on runs from other institutions. For TREC2005 the data are still anonymized, so I was unable to do this, so Nuggeteer may seem to perform well on some runs only because similar runs *are* in the training data.[5]

I report correlation (R^2), and Kendall's τ_b,[6] following Lin and Demner-Fushman. We compute τ against 'corrected' official scores, where official scores are recomputed to assign a nugget to a response if that response was assigned that nugget in any assessment (to correct for inconsistent assessor judgements).

Because Nuggeteer's scores are in the same range as real system scores, I can also report average root mean squared error from the official results.

I varied a number of parameters: stemming, n-gram size, use of judgements for classifier training vs. use of only nugget descriptions (as Pourpre does), *idf* weights vs. count weights, and the effect of removing

[5] I omitted RUN1 for relationship evaluation because it was withdrawn, and because it was identical to RUN10.

[6] Kendall's τ_b is a measure of the similarity of two ranked lists, with values between -1 (reverse order) and 1 (same order).

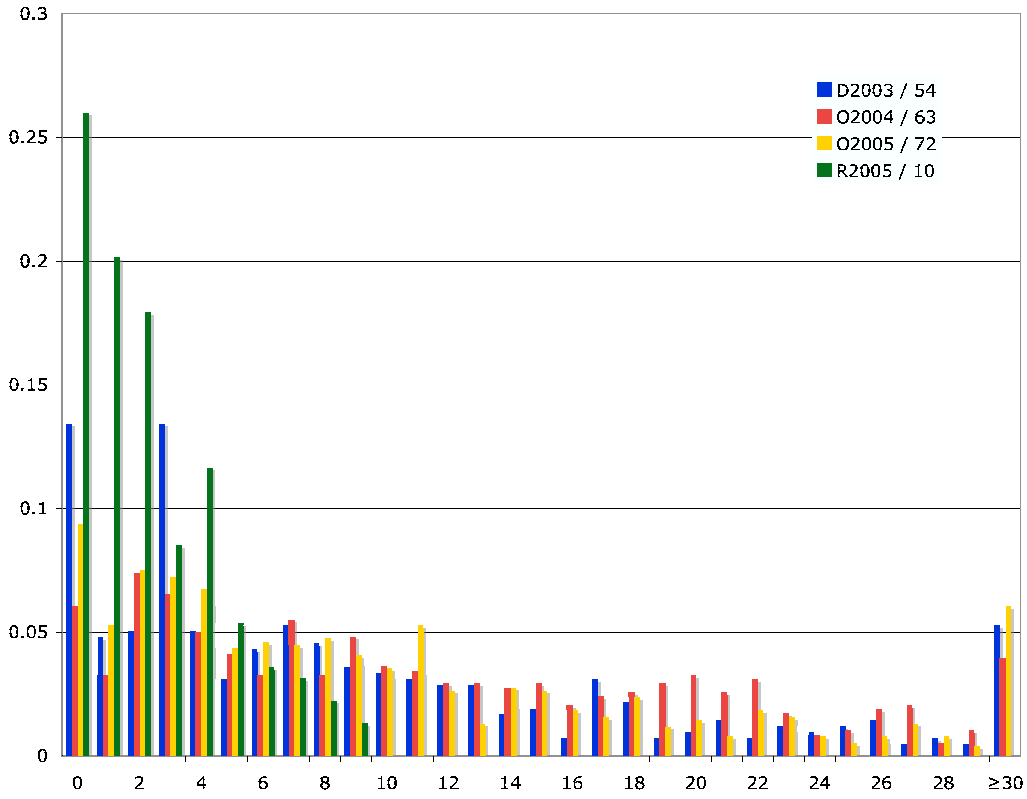


Figure 4: Percents of nuggets, binned by the number of systems that found each nugget.

stopwords. For each experiment, I chose the best performing threshold.

4.1 Comparison with Pourpre

Lin *et al.* report Pourpre and Rouge[7] performance with Pourpre optimal thresholds for TREC definition questions from 2003 through 2005, as reproduced in Table 2 [2]. Comparison with Nuggeteer appears in Table 3.[8]

[7] ROUGE is a text similarity metric commonly used in the summarization literature.

[8] I report only micro-averaged results, because I 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.

Run	POURPRE				ROUGE	
	micro, cnt	macro, cnt	micro, *idf*	macro, *idf*	default	stop
2005 "other" ($\beta = 3$)	0.598	**0.709**	0.679	0.698	0.662	0.670
2004 "other" ($\beta = 3$)	0.785	**0.833**	0.806	0.812	0.780	0.786
2003 "definition" ($\beta = 3$)	0.846	**0.886**	0.848	0.876	0.780	0.816
2003 "definition" ($\beta = 5$)	0.890	**0.878**	0.859	0.875	0.807	0.843

Table 2: Kendall's τ correlation between rankings generated by Pourpre/Rouge and official scores.

Run	POURPRE	ROUGE	NUGGETEER
	macro, cnt	stop	nostem, bigram, micro, *idf*, stop
2005 "relationship" ($\beta = 3$)	0.697		1
2005 "other" ($\beta = 3$)	0.709	0.670	0.906
2004 "other" ($\beta = 3$)	0.833	0.786	0.885
2003 "definition" ($\beta = 3$)	0.886	0.816	0.864
2003 "definition" ($\beta = 5$)	0.878	0.843	0.871

Table 3: Kendall's τ between rankings generated by Pourpre/Rouge/Nuggeteer and official scores. For nuggeteer, all scores reflect the best threshold given no stemmin, bigrams learned, *idf* weights, stopwords removed, and microaveraging.

Table 4 shows a comparison of Pourpre and Nuggeteer's correlations with official scores. As expected 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 [5].

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 I echo Lin and Demner-Fushman's observation that higher correlation scores may reflect overtraining rather than improvement in the metric.

Accordingly, rather than reporting the best Nuggeteer scores (Kendall's τ and R^2) above, I 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 particular thresholds for each year are experimentally optimized.

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

Table 4: 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 in Table 3.

A scatter plot of Nuggeteer performance on the definition tasks is shown in Figure 5.

4.2 Training on Judgements

Judgements are always used, even when we "train only on descriptions" when a system response is identical to a judged response[9]. The distinction for this section is whether we include n-gram features from the judgements in the classifier. Intuitively, if a nugget of information is expressed in a system response, then another response with similar n-grams may also express the same nugget of information.

Unfortunately, the assessors do not mark which *portion* of the response expresses the nugget in question; therefore these n-grams also yield spurious similarity, as shown in Figure 6.

The best results with learning n-grams from judgements for definition questions from 2003 were comparable to the Rouge scores in Figure 2: $\tau = 0.840$, $R^2 = .959$, $\sqrt{mse} = .067$ for the same case as above: no stemming, bigram features, *idf* weighting, keeping stopwords.

4.3 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. I 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 7 compares Kendall's τ over the possible thresholds, n-gram lengths, and stemming. The choice of threshold matters by far the most.

[9] aside from normalizing spaces and case

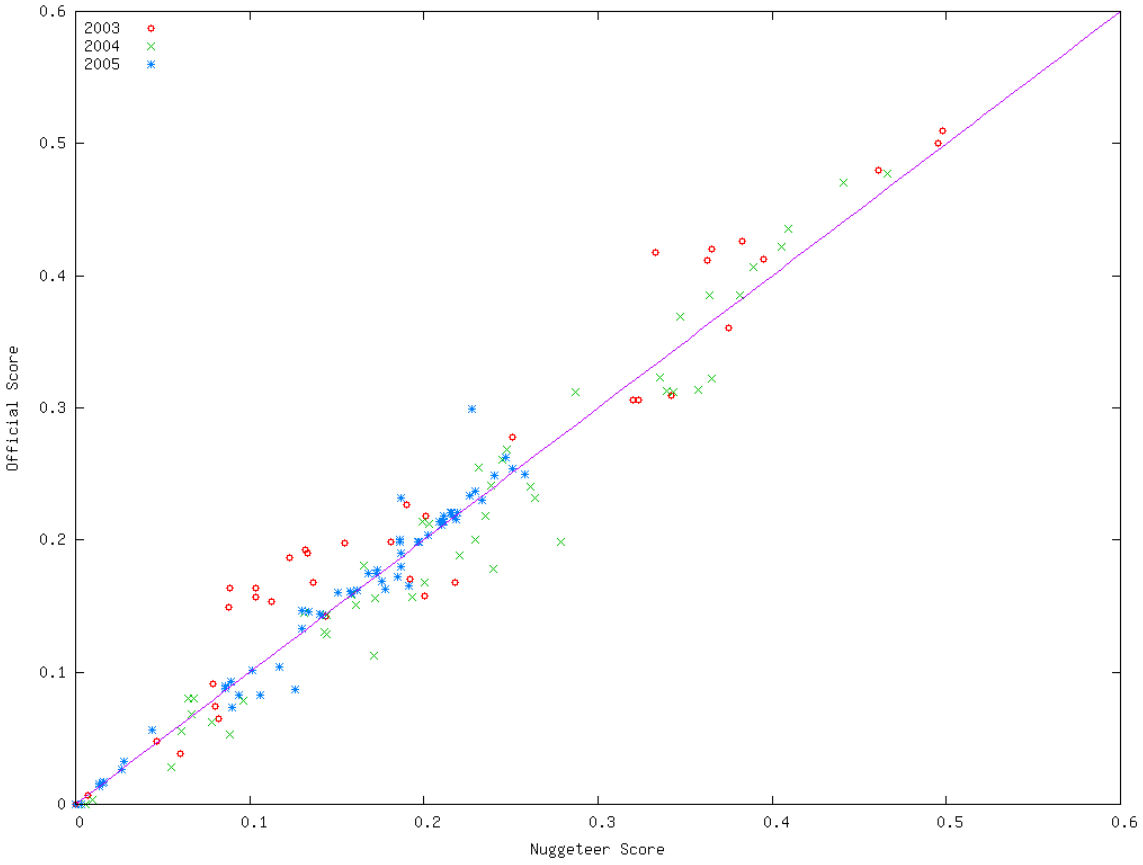


Figure 5: Scatter graph of official scores plotted against Nuggeteer scores (*idf* term weighting, no stemming, bigrams) for three years of definition questions with F($\beta = 3$).

4.4 Term weighting and stopwords

Removing stopwords significantly hurt precision among description-only runs because many of the descriptions were now so short that recall became more coarse-grained, and thus more difficult to threshold. Count weighting (where $W(g, w_1, ..., w_n) = n$) also hurt precision because now common words got too much credit, even when stopwords were removed. Consider: with count weighting, the nugget "American Communist", describing Aaron Copland, has a bigram recall of $^1/_3$ for any response containing the word "American".

The highest count-weighted τ on data from 2003 is 0.503 with bigrams and without stemming. The highest stopwords-removed run has τ 0.320 with unstemmed unigrams. Similar results, though with smaller differences, were obtained when n-gram learning from judgements was enabled.

question id: 1901

response rank: 2

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)

score: 0.14

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 6: 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.

4.5 Interpretability and Novel Judgements

I have shown that Nuggeteer's scores are interpretable by showing root mean squared error less than the measured annotator error. More powerfully, I can show that Nuggeteer's 95% confidence intervals, which it produces on its scores by microaveraging results over questions, are also useful.

Figure 8 shows these confidence intervals. Unsurprisingly Nuggeteer's prediction is better where the correlation with official scores is better.[10]

Unfortunately, the one run from 2005 whose value was outside its predicted confidence interval was, in fact, the best run (RUN-80). This illustrates the problem that without human examination, new and possibly better results may be undervalued by any method that relies on similarity with known answers.

To help overcome this inherent difficulty, Nuggeteer provides its guesses in the assessment format. Human annotators (presumably the developers) may then make judgements about the particular responses returned by the system and add those to the judgement pool. In so doing, they will provide human-level judgements for the particular responses they have judged.

As an example, I manually reassessed RUN-80 and compared my judgements against the assessors' judgements. Of the 597 responses, 6 were known correct and 15 known incorrect from other systems.

I started with a low-threshold set of Nuggeteer judgements, producing a high false positive rate and a

[10] A cautionary reminder: 2005 predictiveness may be artificially high because evaluations were not blind to other submissions from the same group.

lower false negative rate, because I found that false positives were easier to assess. This is clearly not how the assessors should operate, but it is a reasonable and expedient scenario for developers.

Assessing took about 6 hours. It is not very difficult when one already has an answer key, but I was often tempted to amend that answer key to add central facts that RUN-80 included. During assessment, I felt that I was being too strict. My perception is that assessing a run from scratch—without the aid of the nugget descriptions—as the assessors must do, is a tremendously difficult, and perhaps a somewhat arbitrary, task.

Compared with the assessors, my F-measure (based on precision and recall of nugget assignments) was 0.803 ± 0.065. This is significantly higher than Nuggeteer's on its own: 0.503 ± 0.084. My precision was close to 78%, while my recall was 90%. Because I was comparatively too generous in assigning nuggets, RUN-80's final score given my judgements was $.346 \pm .07$. The official score of .299 is within my new confidence interval, whereas it was not within Nuggeteer's original estimate $.227 \pm .06$.

I later learned that RUN-80 was a manual run, produced by a student at the University of Maryland, College Park, working with Jimmy Lin.

5 Discussion

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

- an interpretable score, comparable to official scores,
- a confidence interval on the estimated score,
- scoring known responses exactly,
- support for improving the accuracy of the score through additional annotation, and
- support for using judgement data—not only nugget descriptions—in training.

I have shown that Nuggeteer evaluates the definition and relationship tasks with comparable rank swap rates to Pourpre. I explored the effects of stemming and term weighting, and found that for Nuggeteer, stemming did not help, and that *idf* term weighting was better than count-based. I explored the effects of varying n-gram size and stopword removal, though neither had a great impact.

My exploration of the use of judgement data in training language models for each nugget points 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.

6 Acknowledgements

I would like to thank Jimmy Lin and Dina Demner-Fushman for valuable discussions, for help with the text, and for providing the pioneering baseline. Thanks to Ozlem Uzuner for valuable comments on earlier drafts of this paper. Finally, I am grateful to Boris Katz for his inspiration and unwavering support.

References

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

[2] Jimmy Lin, Eileen Abels, Dina Demner-Fushman, Douglas W. Oard, Philip Wu, and Yejun Wu. A menagerie of tracks at Maryland: HARD, Enterprise, QA, and Genomics, oh my! In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, November 2005.

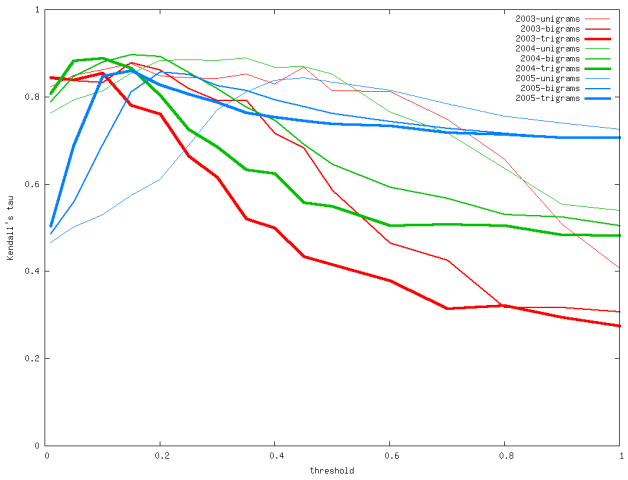
[3] Jimmy Lin and Dina Demner-Fushman. Automatically evaluating answers to definition questions. LAMP 119, University of Maryland, College Park, February 2005.

[4] Jimmy Lin and Dina Demner-Fushman. Will pyramids built of nuggets topple over? LAMP 127, University of Maryland, College Park, December 2005.

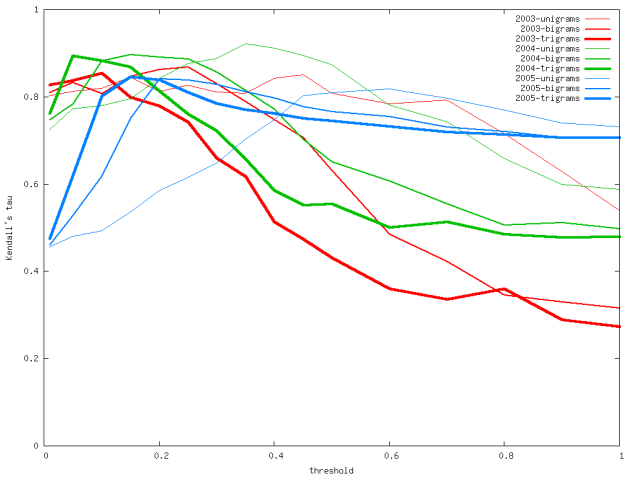
[5] Ellen Voorhees. Overview of the TREC 2003 question answering track, 2003.

[6] Ellen Voorhees. Overview of the TREC 2004 question answering track, 2004.

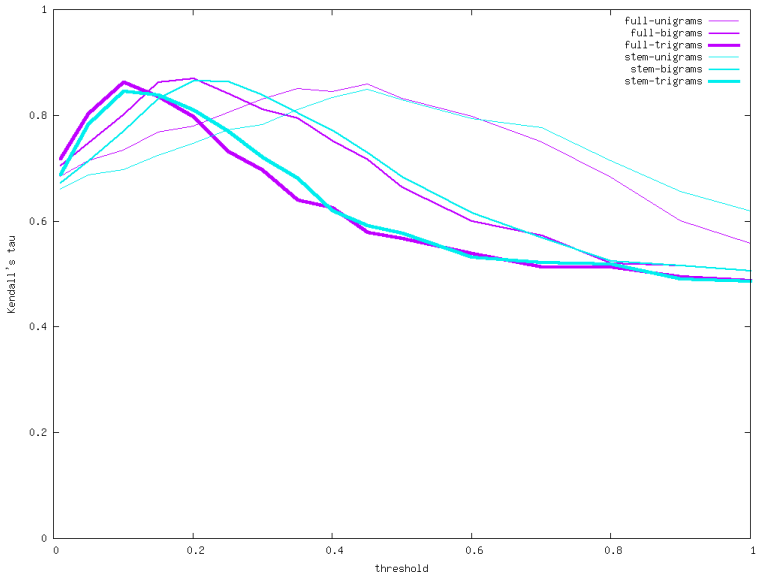
[7] Ellen Voorhees. Overview of the TREC 2005 question answering track, 2005.



(a) unstemmed

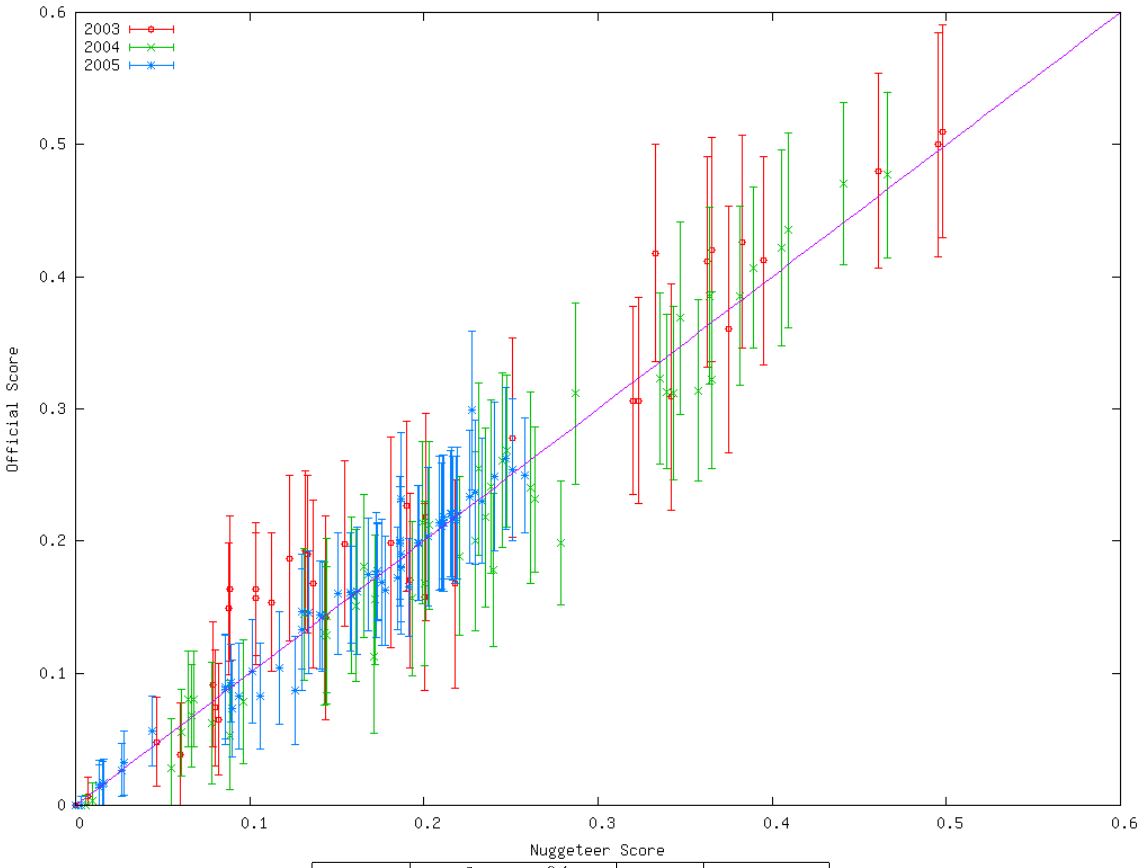


(b) stemmed



(c) comparison of averages

Figure 7: Fixed thresholds vs. Kendall's τ for unigrams, bigrams, or trigrams at $F(\beta = 3)$ on 2003, 2004, and 2005 definition data (a) without stemming (b) with stemming and (c) averaged over the three years to compare unstemmed (purple) and stemmed (blue). All curves peak at roughly the same performance, though the length of the n-gram influences the peak threshold: these factors make little difference in overall performance.



Year	within 95% ci	runs	percent
2003:	44	54	82%
2004:	59	63	94%
2005:	71	72	99%
all:	158	189	92%

Figure 8: Official scores vs. Nuggeteer scores (same as Figure 5) with confidence intervals added. The table shows how many official scores were within the 95% confidence interval of the predicted Nuggeteer scores for each year. The actual confidence appears to be somewhat less than 95%.