Identifying Nuggets of Information in GALE Distillation Evaluation

Olga Babko-Malaya, Greg Milette, Michael Schneider, Sarah Scogin

BAE Systems
6 New England Executive Park Burlington MA 01803 USA
olga.babko-malaya/greg.milette/michael.k.schneider /sarah.scogin@baesystems.com

Abstract

This paper describes an approach to automatic nuggetization and implemented system employed in GALE Distillation evaluation to measure the information content of text returned in response to an open-ended question. The system identifies nuggets, or atomic units of information, categorizes them according to their semantic type, and selects different types of nuggets depending on the type of the question. We further show how this approach addresses the main challenges for using automatic nuggetization for QA evaluation: the variability of relevant nuggets and their dependence on the question. Specifically, we propose a template-based approach to nuggetization, where different semantic categories of nuggets are extracted dependent on the template of a question. During evaluation, human annotators judge each snippet returned in response to a query as relevant or irrelevant, whereas automatic template-based nuggetization is further used to identify the semantic units of information that people would have selected as 'relevant' or 'irrelevant' nuggets for a given query. Finally, the paper presents the performance results of the nuggetization system which compare the number of automatically generated nuggets and human nuggets and show that our automatic nuggetization is consistent with human judgments.

Keywords: machine translation evaluation, question answering evaluation, nuggets of information

1. Introduction[1]

Quantitative evaluations of question answering and summarization, such as TREC QA and DUC evaluations conducted by NIST, employ an evaluation methodology where humans are asked to identify fundamental units of information, called nuggets (Voorhees, 2003) or summary content units (SCUs) (Nenkova and Passonneau, 2004). However, manual annotation is time consuming and limits the volume of responses to be evaluated, having an impact on the statistical significance of the results. Furthermore, several papers have raised the question of whether human-based nugget annotations are stable and whether it is possible to extract information units consistently (e.g. Lin and Zhang, 2007).

Computational approaches to extraction of nuggets (e.g. Marton and Radul, 2006; Zhou et al, 2007) do not face the consistency problem and can process large volume of data. However, a challenge for automatic nuggetization is that the choice and the granularity of the nuggets can vary dependent on the question or topic. For example, when people are selecting nuggets in question answering evaluations, they often choose different units of information dependent on the question they are evaluating (Babko-Malaya, 2008).

[1] This material is based upon work supported by the Defense Advanced Research Projects Agency DARPA/IPTO, Global Autonomous Language Exploitation, contract #HR0011-06-C-003. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The paper is approved for Public Release, Distribution Unlimited.

In this paper, we present an approach to automatic extraction of nuggets employed in the DARPA GALE Distillation evaluation which exploits the fact that different types of questions expect different answers. In this approach, nuggets are categorized according to their semantic type and different semantic categories of nuggets are extracted dependent on the type of the question. The paper describes our approach to nuggetization, the implemented system, and the performance results, which show that our automatic nuggetization is consistent with human judgments.

2. GALE Distillation Evaluation

The goal of GALE Distillation is to return information extracted from multiple source types and languages in response to an open-ended query. The queries conform to templates, which contain argument variables that range over events, topics, people, organizations, locations, and dates, such as DESCRIBE THE ACTIONS OF [person] DURING [date] TO [date]. GALE engines distill data from audio and text sources in multiple languages and produce English-only snippets in response to these queries, which may consist of exact text extractions, translations, summarizations, or paraphrases of the source material. These output responses should contain relevant and non-redundant information: systems are penalized for returning irrelevant and redundant snippets.

The main goal of the evaluation is to quantify the amount of relevant and non-redundant information a distillation engine is able to produce in response to a specific query, and to compare it to the amount of information gathered by a bilingual human using commonly available

state-of-the-art tools.

During evaluation, human annotators judge each snippet as relevant or irrelevant to the query, as well as identify redundant snippets. In order to quantify the amount of relevant, irrelevant, and redundant information and to compare that information across different distillers, we decompose snippets into nuggets or atomic units of information and score distiller outputs at the nugget level. In Phases 1 and 2 of the program, annotators manually parsed all relevant responses into nuggets, using nuggetization annotation guidelines (see Babko-Malaya, 2008), whereas irrelevant nuggets were estimated based on character counts (see White et al, 2008). In later phases, the amount of queries to be evaluated significantly increased in order to ensure statistical significance of the results across languages and various conditions (such as text vs. audio, and newswire vs. blogs). An automatic nuggetization system developed in Phase 4, combined with some improvements in the annotation pipeline, allowed us to process a significantly larger volume of snippets, from more than 20 times more queries. Another advantage of the automatic system is that both relevant and irrelevant snippets are nuggetized, which results in a more meaningful comparison of relevant and irrelevant information.

The scoring system computes the volume of nuggets in relevant, irrelevant, and redundant snippets provided by the distiller and calculates precision and recall scores. Precision scores are computed as the ratio between the number of right nuggets (i.e. relevant and non-redundant) and the number of all nuggets retrieved by a given distiller for a given query, whereas recall is the ratio of the number of right nuggets and the total (estimated) number of non-redundant relevant nuggets for a given query. Because of the large size of the corpus, no Gold Standard was created. Furthermore, unlike TREC evaluations, where an answer key was created by using responses as well as research performed during the original development of the question, GALE Distillation evaluation is only using the pool of responses produced by machine and human distillers.

3. Nuggets in GALE Distillation Evaluation

3.1 Manual Nuggetization

The outputs produced by GALE distillation systems are not restricted to exact answers and may consist of exact text extractions, translations, summarizations, or paraphrases of the source material. As a result, snippets returned by different systems often contain different amount of relevant information, even when this information is coming from the same source and the same sentence. Furthermore, system responses often vary because of machine translation, transcription, as well as co-reference errors.

In the example below, two systems returned snippets for the query PROVIDE INFORMATION ON [Jack Straw]. Distiller 1 returned Snippet 1, where all information is correctly translated. The snippet returned by Distiller 2, on the other hand, contains correct information that Jack Straw went to a summit, but does not contain the facts that Jack Straw was in Egypt and that he is a British Foreign Secretary, which are relevant to the query (the system either did not return the whole sentence or incorrectly translated a part of the sentence).

Snippet 1 (distiller 1): British Foreign Secretary Jack Straw went to a summit in Egypt

Snippet 2 (distiller 2): Snippet: Jack Straw went to a summit ...

As this example shows, in order to compare the volume of correct information returned by distillers, snippets have to be broken down into smaller units, such as nuggets of information, and evaluation needs to compare system responses at the nugget level rather than at the snippet level.

There is a large number of ways to break a sentence down into smaller pieces of information. In Phases 1 and 2 of the GALE program, we developed a manual approach to the creation of nuggets, based on a small set of predefined rules (see Babko-Malaya, 2008), which include the following ones:

- Nuggets are created out of each core verb and its arguments
- Temporal, locative, causative modifiers constitute a nugget
- Numerical expressions, people, organizations, GPEs and titles make a nugget

Given these rules, the following four nuggets are created for Snippet 1 (the extent of the nuggets is indicated by double brackets), whereas only one nugget (which is equivalent to Nugget 3 below) is created for Snippet 2.

> *Nugget1 (person). British Foreign Secretary [[Jack Straw]] went to a summit in Egypt*
>
> *Nugget2 (title). [[British Foreign Secretary]] Jack Straw went to a summit in Egypt*
>
> *Nugget3 (event). British Foreign Secretary [[Jack Straw went to a summit]] in Egypt*
>
> *Nugget4 (location). British Foreign Secretary Jack Straw went to a summit [[in Egypt]]*

3.2. Template-based Nuggetization

A challenge for an automatic approach to nuggetization is the dependence of relevant nuggets on the query. When human annotators decompose a snippet into nuggets, not all of possible nuggets are actually generated. For

example, different relevant nuggets are generated from Snippet 1 for the following queries:

WHICH people ARE INVOLVED IN [summit in Egypt]:
Nugget1 (person). British Foreign Secretary [[Jack Straw]] went to a summit in Egypt
Nugget2 (title). [[British Foreign Secretary]] Jack Straw went to a summit in Egypt

WHERE HAS [Jack Straw] BEEN AND WHEN?,
Nugget4 (location). British Foreign Secretary Jack Straw went to a summit [[in Egypt]]

PROVIDE INFORMATION ON [Jack Straw]
Nugget2 (title). [[British Foreign Secretary]] Jack Straw went to a summit in Egypt
Nugget3 (event). British Foreign Secretary [[Jack Straw went to a summit]] in Egypt
Nugget4 (location). British Foreign Secretary Jack Straw went to a summit [[in Egypt]]

Our automatic approach to nuggetization accounts for this dependency of nuggets to the query by (1) categorizing nuggets into semantic types, and (2) restricting nuggets for each question to a predefined set of semantic categories, as shown in Table 1:

QUERY TEMPLATE	PER	GPE	ORG	TITLE	NUM	EVT	TMP	LOC	MOD	STM
List facts about [EVT]					x	x	x	x	x	x
What [PER/ORG/GPE] are involved in [EVT]?	x	x	x	x						
Provide information on[PER].				x	x	x	x	x	x	x
Find statements made by [PER] on [topic].							x	x	x	x
Describe the relationship of [PER] TO [PER] .						x	x	x		
How did [country] react to [EVT]?					x	x	x	x	x	x
Find acquaintances of [PER]?	x									
Find people who visited [LOC].	x									
List locations of representatives of [ORG/GPE]								x		
Describe a meeting or contact between [PER/ORG] and {PER/ORG]						x	x	x	x	x
Where has [PER] been and when?							x	x		
Which sources made statements on [topic]?	x	x	x	x						

Table 1. Nugget categories for different templates

The semantic categories are defined so that they can be identified consistently given available NLP tools. These categories include PER (person), GPE (geo-political entity), ORG (organization), TITLE (titles), NUM (numerical expressions), EVT (propositional or 'core' nuggets, these nuggets are formed by the verb and its arguments), TMP (temporal expressions), LOC (locative expressions), MOD (other types of modifiers, such as causative, purpose, manner, recipient), and STM (statement nuggets, which indicate direct or indirect speech).

3.3. Scoring with Automatically Generated Nuggets

Prior to automatically generating nuggets, annotators manually tag each snippet as relevant or irrelevant and in the case of relevant snippets, select the portion of the snippet which is relevant to the query. When selecting relevant text, annotators exclude text which is incorrectly translated or garbled, or is otherwise irrelevant to the query.

Snippets are tagged as relevant if they contain relevant and correctly translated material, however, systems are not required to identify the exact answers. As part of relevancy annotation, bilingual annotators verify machine translation and break each relevant snippet down into relevant text and context, where context includes text that is not directly relevant to the query or is incorrectly translated. This text is not nuggetized, i.e. it is ignored in the evaluation. As a result, there is no penalty for returning additional text beyond the exact answer, as long as the snippet contains some relevant material[2].

For example, the selected relevant text in the snippet below includes "Secretary Rice visited New Delhi" (shown in bold), whereas text "Menon said" is excluded to make sure that no credit is given for the other person name mentioned in the snippet:

FIND PEOPLE WHO VISITED [New Delhi]
Snippet. **Secretary Rice visited New Delhi**, *Menon said.*

The nuggetization system automatically nuggetizes the selected relevant text in relevant snippets (these nuggets count as correct), as well as nuggetizes all text in irrelevant and redundant snippets (the nuggets in these

[2] In order to avoid systems returning large snippets, the corpus was segmented at the sentence level and snippets were limited to one segment.

snippets count as incorrect).

The template-based approach to nugget selection described above then allows us to automatically identify the units of information that people would have selected as 'relevant' or 'irrelevant' nuggets for a given query. For example, given the query below, a system returned two snippets: one relevant and one irrelevant. Annotators scored the first snippet as relevant and selected the whole sentence as relevant text. The second snippet was judged as irrelevant. Based on the template of the query (see Table 1 above), the categories of nuggets that were generated for these snippets included times and locations:

WHERE HAS [Chen Yunlin] BEEN AND WHEN?

Relevant: **Chen Yunlin visited [[Taiwan]]-LOC [[in January]]-TMP** *(2 relevant nuggets)*

Irrelevant: Chen's deputy, Zhang Mingqing, was attacked by protesters during an informal visit [[to the southern Taiwanese city of Tainan]]-LOC *(1 irrelevant nugget)*

Since there are two relevant nuggets and one irrelevant one returned by a system, the precision score for this example is 2/3.

By using automatically generated nuggets as opposed to human nuggets, we significantly increased the speed of annotation to support a larger volume of queries, over 20 times more than in earlier phases. Furthermore, this approach allowed a more meaningful comparison of relevant and irrelevant information compared to earlier phases of the program, where irrelevant nuggets were estimated based on character counts. In the example above, the only pieces of information which were considered and counted as relevant vs. irrelevant were times and locations, the rest of the sentence was ignored in both relevant and irrelevant snippets.

Manual or automatic nuggetization is required because Distillation systems do not need to identify the exact answers and nuggetization is not part of the systems. The main challenges for Distillation systems are to identify relevancy and redundancy of information, in addition to machine translation, automatic speech recognition, and co-reference. Whereas all these critical components of Distillation systems are evaluated manually, automatic nuggetization is used to quantify the volume of correct and incorrect information in order to enable meaningful comparison across distillers.

4. Automatic Nuggetizer System

The nuggetizer creates nuggets using a pipeline of NLP tools, post-processing rules, and template-based customization. The NLP tools include a Named Entity tagger (NE), Semantic Role Labeler (SRL) and Part of Speech (POS) tagger. The post-processing rules are applied to the outputs from the tools and aim to resolve conflicts and improve accuracy. The end result is an automatic nuggetizer that approximates human nuggetization and has higher accuracy than the raw outputs of the tools alone. Table 2 describes the tools necessary to generate the nuggets.

NUGGET TYPE	TOOLS
GPE	NE, SRL, and POS tagger
LOC	NE, SRL, and POS tagger
PER	NE and POS tagger
ORG	NE and POS tagger
TITLE	BAE Title detector
EVT	SRL and POS tagger
STM	BAE Statement detector
MOD	SRL
NUM	SRL and BAE Number detector
TMP	SRL and BAE Date/Time detector

Table 2. NLP tools for each nugget type.

Our NE tagger utilizes MaxEnt training from MALLET (McCallum, 2002). For the Semantic Role Labeler, we have been using open source SRL ASSERT (Pradhan et al, 2004). Table 3 shows the mapping between the SRL labels and the nuggets.

SRL LABELS	**NUGGET TYPE**
ArgM-LOC, ArgM-DIR	LOC
ArgM-TMP	TMP
ArgM-CAU, PRP, MNR, ADV	MOD
ArgM-EXT	NUM
MOD, DIS, ArgM, REC, PRD	No nuggets

Table 3. Mapping between nuggets and SRL labels

Locative (LOC) nuggets are derived from phrases that are tagged as ArgM-LOC or ArgM-DIR (directional phrases) by the SRL, combined with the locations selected by the NE tagger. We noticed that our NE tagger has higher precision in identifying locations than ArgM-LOC phrases, but, on the other hand, SRL is better in defining the correct extent of locations. For example, the NE tagger often splits multiword locations into several nuggets, e.g. a phrase *Paris, France* has two locative nuggets: *Paris* and *France*. The post-processing rules for locations, therefore, include:

- Override SRL nuggets with the NE tagger
- If overlap, use the extent of the SRL LOC
- Merge adjacent LOC nuggets
- Do not generate LOC nuggets embedded in NPs

The purpose of the last rule is to limit locations to those which modify events, for example, *US* was not selected as a LOC nugget in *The US president visited Italy-LOC.*

Geo-political entities (GPE) nuggets are also generated by using SRL and NE. GPE nuggets differ from locations in that they function as agents or patients in the sentence, as in *[[Moscow]]-GPE confirmed that samples were delivered to a laboratory*. Since agents and patients are tagged by SRL as Arg0 and Arg1, we use SRL argument labels to distinguish between GPE and LOC:

- Change NE LOC that is Arg0 or Arg1 to a GPE nugget
- Change any NE GPE that are also Arg2-Arg5 or ArgM-LOC/DIR to a LOC nugget

Person (PER) and Organization (ORG) nuggets are generated by the NE tagger. Post-processing rules restrict these nuggets to the arguments of the verbs in the sentence. For example, PER and ORG nuggets are not generated from possessive proper nouns. In the snippet *Bill Clinton's visit to North Korea* was successful, a PER nugget is not created for *Bill Clinton*.

Temporal (TMP) nuggets combine phrases that are tagged as ArgM-TMP with the nuggets identified by the Date/Time Detector, developed at BAE Systems. Our Date/Time detector uses regular expressions to capture the numerous ways dates and times can be expressed, as well as combinations of date/time expressions, including relative time (e.g. yesterday) and specific dates and times (e.g. 30 Mar 2010, 11:00:00 am). The post-processing rules are further aimed to resolve conflicts with the NUM detector, as well as correct the extent of the temporal phrases:

- If ArgM-TMP overlaps with a NUM nugget, choose TMP nugget
- Merge adjacent TMP nuggets

Numerical (NUM) nuggets include ArgM-EXT from SRL and a Number detector developed by BAE Systems. The Number detector identifies exact numbers, phrases representing a specific quantity (e.g. pair, couple, dozen), and words approximating quantities (e.g. several, few, many).

Modifier (MOD) nuggets include all types of modifiers other than temporal and locative expressions. They include causative and purpose phrases (ArgM-CAU and ArgM-PRP), manner adverbials (ArgM-MNR), as well as all modifying expressions that are labeled as ArgM-ADV by the SRL.

Statement (STM) nuggets utilize a Statement detector developed by BAE Systems, which identifies verbs of 'saying' and tags them as STM-nuggets, as in *President Alejandro Toledo [[denied]]-STM Wednesday that terrorism was on the rise*. STM nuggets are restricted to one STM nugget per snippet in the case of the template FIND STATEMENTS MADE BY [person] ON [topic], since different verbs of saying used in the same sentence usually refer to the same statement.

TITLE nuggets utilize a Title detector, developed by BAE Systems, which extracts pre-and post-modifiers of a named person as defined for the ACE data set (ACE 2004). The detector makes use of a hierarchical, dynamic conditional random field (CRF) model (Sutton et al, 2007) to jointly identify the full extent of the person phrase and tag pre-modifier and post-modifier elements.

Event (EVT) nuggets are defined as verbs (SRL relations). However, infinitival verbs, gerunds, and participles are not included as EVT nuggets. EVT nuggets are also restricted for some templates, for example only one EVT nugget is allowed for questions which ask about a relationship between two people or two organizations. As in the case of some other post-processing rules discussed above, these rules aim for high accuracy that approximates human nuggetization without over-generation of nuggets.[3]

5. Performance Results

In order to assess how well our approach to automatically generating nuggets performed, we compared the results of automatic nuggetization to manually-annotated nuggets. Specifically, we evaluated performance on the results of the Arabic queries in the GALE Phase 4 distillation evaluation. For this language, there were 300 queries resulting in 7061 snippets returned from machine distillers. For each snippet, nuggets were extracted automatically and then corrected manually, given the nuggetization annotation guidelines in Phase 2. The final manually corrected nuggets provide a gold standard against which to compare the automatic nuggetizer. When evaluating performance of the nuggetizer, we are primarily interested in determining whether the total numbers of nuggets are reasonably accurate and whether we are biased high or low in the counts. Table 4 shows the number of nuggets generated by the automatic nuggetizer as compared against the gold standard manually corrected nuggets, broken down by structured/unstructured (i.e. newswire vs. blogs) and audio/text. We see that the difference between auto nuggets and human nuggets is reasonably small, and we consistently undergenerated auto nuggets.

[3] Over-generation of nuggets would negatively affect system scores, since during evaluation nuggets in relevant snippets are manually corrected, whereas the number of nuggets in the irrelevant snippets is estimated by using automatic nuggetization.

	Corrected nuggets	Auto nuggets	**Difference**
Str. Audio	3308	2971	**10%**
Str. Text	7766	7700	**1%**
Unstr. Audio	3108	2795	**10%**
Unstr. Text	4944	4780	**3%**
Overall	**19126**	**18246**	**5%**

Table 4. Relative Difference between Gold Standard Manually Corrected and Automatically Generated Nuggets Broken down by Condition

Examining breakdowns by condition, it is clear that performance is worse for audio than text, presumably due to odd grammatical constructions found in spoken language as well as transcription errors that together made snippets more difficult for the nuggetizer's constituent NLP tools to process. But even for audio, the undergeneration is only about 10%. Note that the differences, although small, are statistically significant. Table 5 shows the results of the statistical analysis of the differences between corrected and automatic nuggets generated for each snippet. Viewing each snippet as providing an independent sample of the difference, we estimate the difference as 0.12 ± 0.04 nuggets (with a 95% confidence interval), and the p-value for rejecting the hypothesis of a zero mean difference as $9x10^{-10}$. Thus, we are very confident both that the number of nuggets generated by the automatic nuggetizer is biased, and that the bias on average results in a slight undergeneration of automatic nuggets.

Difference Mean	95% Confidence Interval Around Mean	p-value for rejecting Mean=0
0.12	0.04	9.E-10

Table 5. Statistical Analysis of the Per Snippet Difference between Gold Standard Manually Corrected and Automatically Generated Nuggets

6. Conclusion

This paper presents an approach to nugget extraction which measures information content of system responses taking into account the dependency of relevant nuggets on the question. Such an approach can be applied to many NLP tasks where it is desirable to evaluate at the nugget level. Examining responses at the nugget level is especially relevant when the information content of responses derived from the same source text may be highly variable across systems. This is the case in GALE Distillation because the systems do not simply return snippets of text from the source document but distill responses containing multiple nuggets into English from other source languages.

7. References

ACE 2004 Multilingual Training Corpus, LDC2005T09

Babko-Malaya, O. 2008. Annotation of nuggets and relevance in GALE distillation evaluation. Proceedings LREC 2008 (Linguistic Resources and Evaluation Conference),, Marrakech, Morocco.

Lin J. and P. Zhang. 2007. Deconstructing Nuggets: The Stability and Reliability of Complex Question Answering Evaluation. In Proceedings of the 30th ACM SIGIR Conference, pages 327-334, Amsterdam, the Netherlands

Marton, G. and A. Radul, 2006. Nuggeteer: automatic nugget-based evaluation using description and judgments. In Proceedings of NAACL-HLT 2006

McCallum K. 2002, "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu.

McCallum and W. Li. 2003. “Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons,” in Proceedings of the Seventh Conference on Natural Language Learning (CoNLL),

Nenkova, A. and R. Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. In Proceedings of the Human Language Technology Conference – North American chapter of the Association for Computational Linguistics annual meeting (NAACL-HLT 2004).

Pradhan S.S, W. Ward, K. Hacioglu, J. H. Martin, D. Jurafsky, 2004. Shallow Semantic Parsing using Support Vector Machines in Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting (HLT/NAACL-2004), Boston, MA,.

Sutton, A. McCallum, and K. Rohanimanesh, 2007. “Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data,” Journal of Machine Learning Research, 8, 693-723,.

Voorhees, E. 2003. Overview of the TREC 2003 question answering track. In Proceedings of TREC 2003.

White, J.V., Hunter, D., and Goldstein, J.D. 2008. Statistical evaluation of information distillation systems. Proceedings of LREC 2008 (Linguistic Resources and Evaluation Conference), Marrakech, Morocco, May.

Zhou, L. N. Kwon, and E.H. Hovy. 2007. A Semi-Automated Evaluation Scheme: Automated Nuggetization for Manual Annotation. In Proceedings of the Human Language Technology / North American Association of Computational Linguistics conference (HLT-NAACL 2007). Rochester, NY