

# Text Comparison Using Machine-Generated Nuggets

Liang Zhou

Information Sciences Institute  
University of Southern California  
4676 Admiralty Way  
Marina del Rey, CA 90292  
liangz@isi.edu

## Abstract

This paper describes a novel text comparison environment that facilitates text comparison administered through assessing and aggregating information nuggets automatically created and extracted from the texts in question. Our goal in designing such a tool is to enable and improve automatic nugget creation and present its application for evaluations of various natural language processing tasks. During our demonstration at HLT, new users will be able to experience first hand text analysis can be fun, enjoyable, and interesting using system-created nuggets.

## 1 Introduction

In many natural language processing (NLP) tasks, such as question answering (QA), summarization, etc., we are faced with the problem of determining the appropriate granularity level for information units in order to conduct appropriate and effective evaluations. Most commonly, we use sentences to model individual pieces of information. However, more and more NLP applications require us to define text units smaller than sentences, essentially decomposing sentences into a collection of phrases. Each phrase carries an independent piece of information that can be used as a standalone unit. These finer-grained information units are usually referred to as *nuggets*.

Previous work shows that humans can create nuggets in a relatively straightforward fashion. A

serious problem in manual nugget creation is the inconsistency in human decisions (Lin and Hovy, 2003). The same nugget will not be marked consistently with the same words when sentences containing multiple instances of it are presented to human annotators. And if the annotation is performed over an extended period of time, the consistency is even lower.

Given concerns over these issues, we have set out to design an evaluation toolkit to address three tasks in particular: 1) provide a consistent definition of what a nugget is; 2) automate the nugget extraction process systematically; and 3) utilize automatically extracted nuggets for text comparison and aggregation.

The idea of using semantic equivalent nuggets to compare texts is not new. QA and summarization evaluations (Lin and Demner-Fushman, 2005; Nenkova and Passonneau, 2004) have been carried out by using a set of manually created nuggets and the comparison procedure itself is either automatic using n-gram overlap counting or manually performed. We envisage the nuggetization process being automated and nugget comparison and aggregation being performed by humans. It's crucial to still involve humans in the process because recognizing semantic equivalent text units is not a trivial task. In addition, since nuggets are system-produced and can be imperfect, annotators are allowed to reject and re-create them. We provide easy-to-use editing functionalities that allow manual overrides. Record keeping on edits over erroneous nuggets is conducted in the background so that further improvements can be made for nugget extraction.

## 2 Nugget Definition

Based on our manual analysis and computational modeling of nuggets, we define them as follows:

Definition:

- A nugget is predicated on either an *event* or an *entity*.
- Each nugget consists of two parts: the anchor and the content.

The anchor is either:

- the head noun of the entity, or
- the head verb of the event, plus the head noun of its associated entity (if more than one entity is attached to the verb, then its subject).

The content is a coherent single piece of information associated with the anchor. Each anchor may have several separate contents. When the nugget contains nested sentences, this definition is applied recursively.

## 3 Nugget Extraction

We use syntactic parse trees produced by the Collins parser (Collins, 1999) to obtain the structural representation of sentences. Nuggets are extracted by identifying subtrees that are descriptions for entities and events. For entities, we examine subtrees headed by “NP”; for events, subtrees headed by “VP” are examined and their corresponding subjects (siblings headed by “NP”) are investigated as possible entity attachments for the verb phrases. Figure 1 shows an example where words in brackets represent corresponding nuggets’ anchors.

## 4 Comparing Texts

When comparing multiple texts, we present the annotator with each text’s sentences along with nuggets extracted from individual sentences (see Appendix A). Annotators can select multiple nuggets from sentences across texts to indicate their semantic equivalence. Equivalent nuggets are grouped into nugget groups. There is a frequency score, the number of texts it appeared in, for each nugget group. We allow annotators to modify the

Sentence:

The girl working at the bookstore in Hollywood talked to the diplomat living in Britain.

Nuggets are:

[girl] working at the bookstore in Hollywood  
[girl] working at the bookstore  
[bookstore] in Hollywood  
girl [talked] to the diplomat living in Britain  
girl [talked] to the diplomat  
[diplomat] living in Britian

Figure 1. Nugget example. (words in brackets are the anchors).

nugget groups’ contents, thus creating a new label (or can be viewed as a super-nugget) for each nugget group. Record keeping is conducted in the background automatically each time a nugget group is created. When the annotator changes the content of a nugget group, it indicates that either the system-extracted nuggets are not perfect or a super-nugget is created for the group (see Appendix B and C). These editing changes are recorded. The recorded information affords us the opportunity to improve the nuggetizer and perform subsequent study phrase-level paraphrasing, text entailment, etc.

## 5 Hardware Requirement

Our toolkit is written in Java and can be run on any machine with the latest Java installed.

## References

- Collins, M. 1999. Head-driven statistical models for natural language processing. *PhD Dissertation*, University of Pennsylvania.
- Lin, C.Y. and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of NAACL-HLT 2003*.
- Lin, J. and D. Demner-Fushman. 2005. Automatically evaluating answers to definition questions. In *Proceedings of HLT-EMNLP 2005*.
- Nenkova, A. and R. Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. In *Proceedings of NAACL-HLT 2004*.