

## INTRODUCTION

Ontology is perhaps the most critical enabling technology in semantic web applications.

An **ontology** basically describes **terms**, and **types of relationship** between each pair of terms. Thus, an ontology can be expressed as a list of tuples in the form of (*term x, relationship r, related term y*). For example, "file server, is a kind of, computer", "memory is a part of computer".

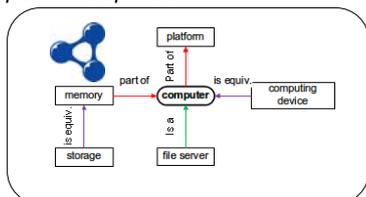


Figure 1, An example ontology from WordNet

## The Development of Ontology

Ontologies are generally developed by a small group of experts, but it is not clear if this is the best approach. Costs can be prohibitive and assembling suitable experts can be time-consuming.

Besides, even experts have difficulty keeping up with advances in knowledge in the open, dynamic World Wide Web environment.

Typical problem tasks in ontology development consist of term selection, relationship assignment, and evolution.

## The Rise of Crowdsourcing

Recently, crowdsourcing has emerged as potentially the most influential way to solve this kind of problems by outsourcing a job traditionally done by experts to non-experts, typically a large group of people in the form of an open call.

## Our Alternative Approach

We explore the feasibility of using non-experts for ontology development. Specifically, we attempt to involve and aggregate knowledge from common web users, such as users from Amazon Mechanical Turk or search engines.

Table 1, Statistics of speed and cost

Domain	HIT	Judgements		Unique Worker	Speed (hours)	Cost (USD)
		Trusted	Untrusted			
Travel	60	525	1135	132	3.5	8.78
Vehicle	60	550	685	62	0.5	15.8
Computer	60	520	450	57	1.2	8.78
Total:	180	1595	2270	251	5.2	33.36



Figure 5, Screenshot of the OntoAssist platform, hahia.com

## ONTOLOGY BUILDING USING MECHANICAL TURK

We first explore the capability of MTurk users for ontology building, both in the task of domain term selection and relationship assignment.

For this purpose, we select a variety of ground-truth ontologies and assess how well web users can reproduce them.

### Task Design

We designated each of the tuples in an ontology as an human intelligence task (HIT). Each HIT was designed to get human's knowledge on a specific term. Thus, the experiment was setup as, given a term x, can the user correctly find the related term y and assign a proper relationship r with the term x.

### Experiments on MTurk

To explore the Turker's capability on different domains, we conducted experiments asynchronously, starting with travel, vehicle, and, finally computer. Each HIT was checked by at least 8 individual Turker.

The result we got was a list of judgments. For instance, a judgment made by a Turker with personal ID 32005449 added a row of data to the result as follow: "32005449, airbus, is less general than, airplane".

### Aggregation

Gold standards has been set up to recognize untrusted users. We then aggregated the trusted results by applying the rule of majority agreement. (See figure 2, 3)

### Results

The experiments completed in a short time at low cost(See Table 1). The results indicate that the crowd achieve an agreement greater than 93% on the recognition of related terms and greater than 48% on the type of relationship between each pair of terms. (See Table 2)

By comparing the results agreed by the majority of workers with the ground-truth ontologies, we observe that workers achieve more than 90% accuracy in the reproduction of the ontologies. (See Table 2)

Table 2, Agreement and Accuracy

Domain	Agreement		Accuracy
	Related term	Relationship	
Vehicle	93.26%	59.21%	95%
Travel	94.78%	48.75%	95%
Computer	97.12%	58.27%	90%

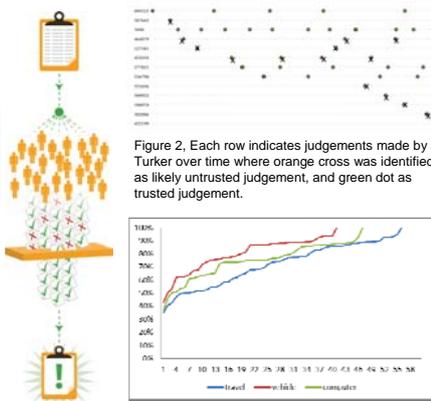


Figure 2, Each row indicates judgements made by a Turker over time where orange cross was identified as likely untrusted judgement, and green dot as trusted judgement.

Figure 3, Agreement for each HIT

## ONTOLOGY EVOLUTION USING SEARCH ENGINE

We also propose OntoAssit, a semantic navigation support tool designed to address the issue of ontology evolution.

We assume that most users are aware of the semantic relationship between the query word and the suggested terms although there is no explicit way for them to express it in most of existing search engines.

### System Architecture

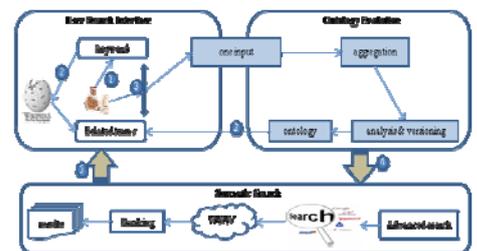


Figure 4, An open crowdsourcing model for ontology evolution with semantic search engine

The ontology evolution model of OntoAssit consists of the following three components.

1) We elicit users' knowledge with related terms generated from Wikipedia. The semantic navigation component allows the users to express their search intent as a tuple (keyword, relationship, related term). For instance, a user can refine an original query python by the tuple: (python, is a kind of, programming language).

2) We then aggregate these terms and relationships from different query sessions. We assume that one expression is correct if majority of the users agree on it.

3) The system will apply the changes periodically to the old version and then release a new one for further editing. Change conflicts will be resolved during the aggregation under majority rule adjusted by user impact.

### Implementation and Evaluation

OntoAssit has been implemented and tested by 225 workers from MTurk in a live demonstration site, www.hahia.com. (See figure 5). Computer ontology, including 87 terms was chosen as test ontology.

We discovered 89% of the 173 new terms were relevant to computer domain. The accuracy of the relationship among them was about 62%. (See figure 6)

Considering that hundreds of billions searches conducted each month, this has the potential to bring a significant change in the way we approach ontology maintenance and evolution, and thus accelerate the maturity of semantic web.

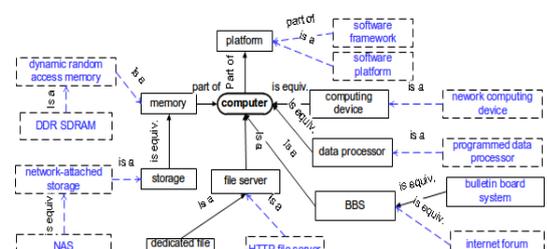


Figure 6, A part of resulting ontological structure. Terms and relationships in blue colour are new collections from users' inputs.