

A Tool for Measuring the Reality of Technology Trends of Interest

Peter Z. Yeh and Colin A. Puri

Accenture Technology Labs
San Jose, CA 95113
{peter.z.yeh,colin.puri}@accenture.com

Abstract

In this paper, we present a prototype application – the Technology Trend Tracker – to measure the reality of technology trends of interest using information on the Web to inform decisions such as when to develop training, when to invest in expertise, and more. This prototype performs this task by integrating several artificial intelligence technologies in an innovative way. These technologies include rich semantic representations, a natural language understanding module, and a flexible semantic matcher. We use our system to augment Accenture’s annual technology vision survey and show how our system performs well on measuring the reality of technology trends from this survey. We also show why our system performs well through an ablation study.

Introduction

Enterprises are continuously exposed to a variety of technology trends from a wide range of media sources such as press releases, trade journals, analyst reports, etc. The challenge for these enterprises is separating the reality (i.e. tangible events such as successful trials, deployments, etc.) from the hype (i.e. buzz, predictions, forecasts, etc.) in order to inform decisions such as when to develop training, when to invest in expertise, and more.

For example, cloud computing is a recent technology trend with much media exposure. Many news stories on this trend are reported each week, but only a few are of tangible events. Hence, many enterprises are paying close attention to this technology trend, but are struggling to understand “how real” it is.

There are solutions available, like the Gartner Hypecycle and Buzz Metrics, that can help enterprises understand the hype around technology trends of interest. These solutions, however, have various limitations. The Gartner Hypecycle is a manual solution. Hence, the resulting assessments are static, capturing the hype around a technology trend at a particular moment in time with infrequent (or no) updates after they are made. Solutions like Buzz Metrics are automated, but they only report on the hype. They neither report on the reality around a technology trend nor provide details on what types of tangible events are occurring (and the amount).

Copyright © 2009, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, we present a prototype application – the Technology Trend Tracker – that complements these existing solutions by automatically tracking various media sources to measure the reality of technology trends of interest. Our system performs this task by integrating the following Artificial Intelligence (AI) technologies:

- Semantic representations to guide the detection of events that indicate the realization of technology trends of interest (i.e. reality events).
- A Natural Language Understanding (NLU) algorithm to capture these representations from news headlines.
- A flexible semantic matcher to detect equivalent events by matching the resulting representations.

We give an overview of the main features of our system followed by a description of its architecture. We use our system to augment Accenture’s annual technology vision survey and show how our system performs well on measuring the reality of technology trends from this survey. We also show why our system performs well through an ablation study.

Application Overview

The Technology Trend Tracker measures the reality of technology trends of interest. It measures this information by tracking media sources on the Web – e.g. RSS feeds, press releases, analyst reports, etc. – and presents the result via an executive dashboard. For each technology trend, this dashboard reports the number of unique reality events for the trend (i.e. reality) over a user-specified time interval. It also reports the number of headlines about the trend (i.e. hype) for comparison purposes (see Figure 1).

The Technology Trend Tracker reports two additional pieces of information to give users a better understanding of the reality of each technology trend. It reports the distribution of reality event types for each trend. Figure 2 shows the distribution of event types such as deployment, alliance, etc. for cloud computing. The user can click on any event type to examine the actual events and the headlines from which they were detected.

The Technology Trend Tracker also reports the most active companies for each trend. It ranks a company by the number of unique reality events performed by the company.

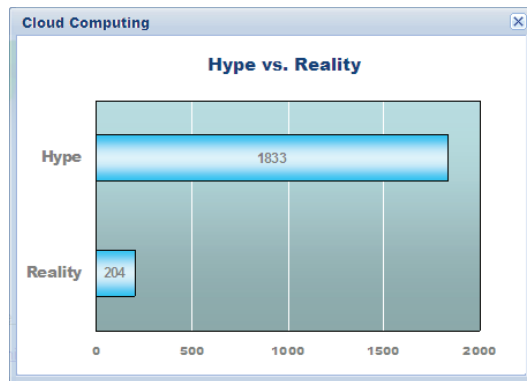


Figure 1: The hype versus reality for the technology trend cloud computing over a six month period.

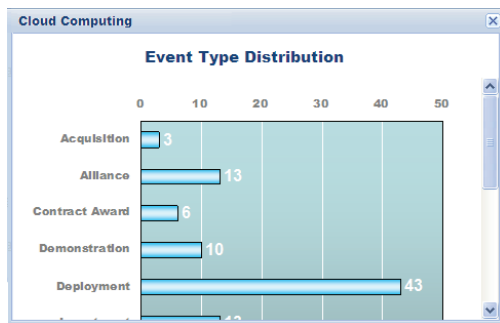


Figure 2: The distribution of different reality event types for cloud computing.

Figure 3 shows the six most active companies for cloud computing. The user can click on a company of interest to examine the events performed by the company.

System Overview

The architecture of the Technology Trend Tracker has four components: semantic representations, a trend filter, a NLU module, and a semantic matcher (see Figure 4). Semantic representations guide the detection of technology trends and events which indicate their realization (i.e. reality events) from various media sources. The trend filter keeps only headlines about technology trends of interest. The NLU module captures semantic representations of reality events from the remaining headlines. The semantic matcher matches the resulting representations to detect equivalent events.

Semantic Representations

Semantic representations are used by the trend filter and the NLU module to detect technology trends and their reality events from media sources. These representations need to provide rich semantics and be linguistically motivated, so we chose the Component Library (CLib) built by Barker et al. (Barker, Porter, and Clark 2001) over other resources

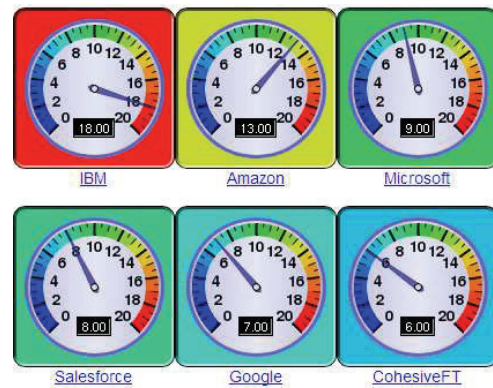


Figure 3: The top six most active companies for cloud computing.

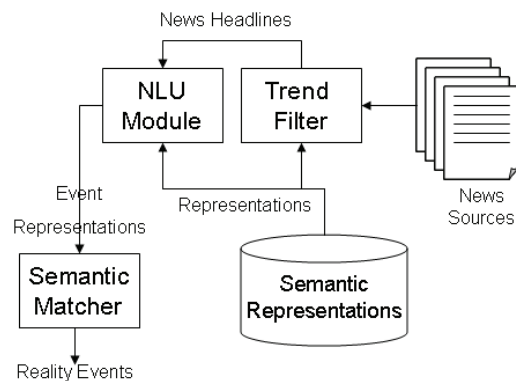


Figure 4: The schematic of our system's architecture.

like WordNet (Fellbaum 1998) and FrameNet (Baker, Fillmore, and Lowe 1998). The semantics of concepts in WordNet are limited mostly to hypernyms, meronyms, and synonyms while FrameNet focuses primarily on the semantic roles played by the syntactic arguments of verbs. The CLib, on the other hand, provides a rich domain independent upper ontology with about 80 semantic relations and about 500 generic concepts that can be composed and extended to build domain specific ones.

The semantic relations fall into three general categories. 1) Relations between an event and an entity such as *agent*, *object*, and *recipient*. These relations are in the spirit of case roles proposed by Fillmore (Fillmore 1971) and others (Barker 1998). 2) Relations between entities such as *has-part*, *possesses*, and *material*. 3) Relations between events such as *caused-by*, *prevents*, and *enables*. Each semantic relation is annotated with information about its syntactic realization (Barker 1998) – i.e. how the relation surfaces in language. For example, *agent* can surface as the subject of a sentence (e.g. “a man hit a ball”), and it can also surface as a prepositional phrase marked by the preposition “by” (e.g. “A ball was hit by a man”). This information is used by the NLU module to capture rich semantic representations from news headlines.

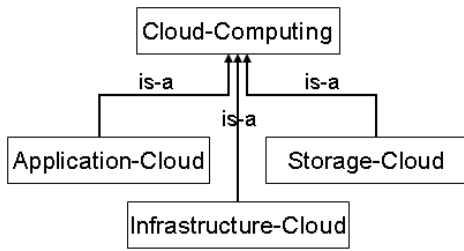


Figure 5: A portion of the is-a hierarchy for the cloud computing technology trend.

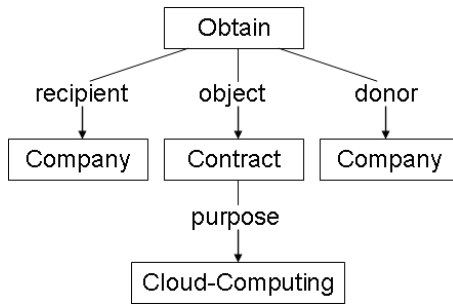


Figure 6: The representation of a reality event: a company awarding a contract for cloud computing to another company. This representation is drawn as a conceptual graph (Sowa 1984).

The concepts in the CLib cover events such as *Purchase*, *Transfer*, etc. and entities such as *Organization*, *Artifact*, etc. Each concept is annotated with appropriate senses from WordNet to provide information about the concept’s lexical realization. For example, the *Purchase* event is annotated with the WordNet senses of *buy#1*, *purchase#1*, etc. This information is used by the trend filter to detect headlines about technology trends of interest and by the NLU module to capture semantic representations from the remaining headlines.

We use these generic relations and concepts to customize our system by encoding technology trends of interest – e.g. cloud computing, see Figure 5 – and events that indicate the realization of these trends – e.g. a company awarding a contract for cloud computing to another company, see Figure 6.

This customization effort took us approximately 4 weeks to complete. We spent the bulk of this effort on interviewing subject matter experts in each trend to identify 1) technologies that make up a trend, 2) how these technologies relate to each other, and 3) events that indicate the realization of the trend.

Trend Filter

The trend filter keeps only news headlines about technology trends of interest. For each headline, this filter looks up the lexical realizations for technology trends encoded in the CLib. It then examines whether these realizations occur in the headline. If so, then this filter counts the headline towards the hype of the trend whose realization occurred in

the headline. Otherwise, the headline is discarded. The remaining headlines are passed to the NLU module which determines whether they report on reality events.

NLU Module

Our system requires a NLU module to capture representations of reality events from the remaining headlines, so we use the approach by (Yeh and Kass 2008) because it has been shown to perform well on this task. We briefly describe how this approach works, and refer the reader to (Yeh and Kass 2008) for a complete discussion along with a full example.

This approach takes the representation of a reality event and generates a set of phrases from it. These phrases are generated by randomly selecting a syntactic realization for each semantic relation (i.e. edge in the representation) and a lexical realization for its value (i.e. the concept pointed to).¹ For example, given the representation from Figure 6 this approach generates the phrase “to company” for the semantic relation *recipient* and its value *Company* because “to” is a syntactic realization for *recipient* and “company” is a lexical realization for *Company*.

This approach then matches these phrases against the subject, syntactic object, and prepositional phrases in each headline. A phrase *p* generated from a representation matches a phrase *q* in a headline if 1) they begin with the same syntactic marker (e.g. “subject”, “by”, etc.) and 2) the stem of the lexical realization in *p* is identical to the stem of a constituent in *q*. If *p* cannot be matched, then a set of syntactic and semantic rewrites is applied to improve the match. For example, one of these rewrites allows the lexical realization for a concept to be replaced with the lexical realization for one of its subclasses – e.g. “transfer” for the concept *Transfer* can be replaced with “buy” for the concept *Purchase*, a subclass of *Transfer*. This matching process repeats until all phrases generated from a representation are matched or rewrites no longer improve the match.

In our system, we require all phrases generated from the representation of a reality event to match a headline in order for the semantics of the headline to be captured by the representation.

Semantic Matcher

The same reality event can be expressed in different ways (see Figure 7) or at different levels of detail (see Figure 8). To prevent over-counting, our system uses a flexible semantic matcher that matches the representations captured by the NLU module to detect equivalent reality events. We give a brief description of this matcher, and point the reader to (Yeh, Porter, and Barker 2003) for additional details.

This semantic matcher takes two representations (encoded as either conceptual graphs (Sowa 1984) or RDF graphs (Lassila and Swick 1999)) and uses taxonomic knowledge (regarding both concepts and relations) to find the largest connected subgraph in one representation that is isomorphic to a subgraph in the other. This matcher then uses a library of over 200 transformation rules to shift

¹Only a lexical realization is selected when a concept is not the value of any semantic relation.

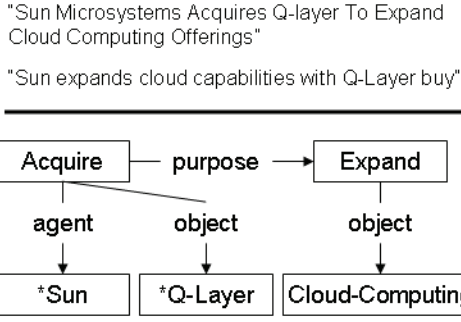


Figure 7: **Top:** Two headlines reporting the same event. **Bottom:** The semantic representation of the event reported by both headlines.

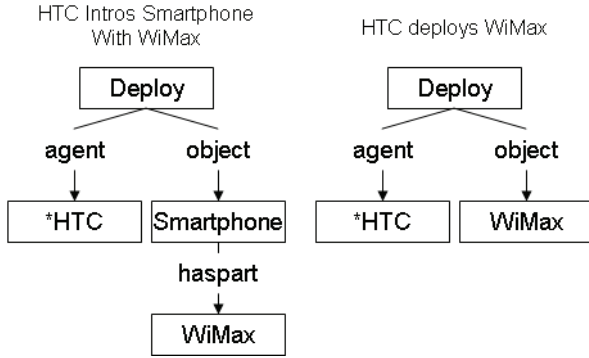


Figure 8: Two headlines reporting the same event at different levels of detail. The representation of each headline is shown below the headline.

the representations to improve the match. This improvement might enable other subgraphs to match isomorphically, which in turn might enable more transformation rules, and so on until the match improves no further.

This library of transformation rules is based on the CLib’s upper ontology and has been used by Yeh et al. to improve matching in the domains of battle space planning (Barker et al. 2003) and office equipment purchasing (Yeh, Porter, and Barker 2005). Each rule is an instance of the pattern “transfers through” (Lenat and Guha 1990), which is shown at the top of Figure 9. C_i is a concept in this pattern, and r_j is a relation. Examples of rules that obey this pattern are shown at the bottom of Figure 9. The first rule encodes part descension (i.e. acting on a whole also acts on its parts), and the second rule encodes the transitivity of the *has-part* relation. A transformation rule is applicable to a representation if its antecedent subsumes a subgraph of the representation. The first rule, for example, can be applied to the left representation in Figure 8 to resolve the mismatch between it and the representation on the right.

We say two representations are equivalent if all the concepts in one representation can be matched to the concepts in the other.

$$C_1 \xrightarrow{r_1} C_2 \xrightarrow{r_2} C_3 \Rightarrow C_1 \xrightarrow{r_1} C_3$$

$$Event_1 \xrightarrow{object} Entity_2 \xrightarrow{haspart} Entity_3 \Rightarrow Event_1 \xrightarrow{object} Entity_3$$

$$Entity_1 \xrightarrow{haspart} Entity_2 \xrightarrow{haspart} Entity_3 \Rightarrow Entity_1 \xrightarrow{haspart} Entity_3$$

Figure 9: **Top:** The “transfers through” pattern. **Bottom:** Two example rules that obey this pattern.

Evaluation

We evaluated the performance of the Technology Trend Tracker on measuring the reality of technology trends from Accenture’s technology vision survey.

Background

Accenture – a global technology consulting company – conducts an annual survey on technology trends of interest to itself and its clients. This survey provides insight into “how real” are these trends based on indicators such as how many vendors have entered the market, the number of deployments, and more. This survey, however, is conducted manually and once a year. Hence, the results can quickly become outdated.

We used our system to provide a continuous measure of the reality of the top 10 trends from this survey. These trends include Business Intelligence (BI), cloud computing, Identity Management (IDM), mashup, mobile platforms, Rich Internet Applications (RIA), Software as a Service (SaaS), Service Oriented Architecture (SOA), social media, and WiMax. We customized our system for this task by extending the CLib (see earlier section titled *Semantic Representations*) to include these technology trends and the events that indicate their realization – there was a total of 38 reality events. We also pointed our system to 350 online news sources to detect these trends and events. These sources ranged from mainstream media venues (e.g. CNN, WSJ, etc) to technology specific ones (e.g. Wired, Technology Today, etc).

Table 1 shows the number of unique reality events reported by our system for these 10 trends from the 350 online sources over a three month period. This table also shows the total number of news headlines about these trends from the same sources over the same time period for comparison purposes.

Experiment Setup and Result

We evaluated the performance of our system based on the precision² of the reality events reported over a three month period – i.e. the number of correct reality events reported by our system over the total number of reality events reported

²We did not evaluate our system’s recall because of limitations in time and resources.

Tech Trend	Hype	Reality
BI	246	42
Cloud	906	95
IDM	79	12
Mashup	55	8
Mobile Platform	1302	97
RIA	199	22
SaaS	518	72
SOA	382	32
Social Media	894	46
WiMax	207	42
Overall	4788	468

Table 1: The hype versus reality reported by our system for 10 technology trends over a three month period.

by our system. We did not evaluate a competing system because one does not exist.

To obtain this metric, we employed two human graders to evaluate the correctness of these events. We gave each grader the headline from which an event was detected and the semantic representation of the event. We also gave each grader the 38 reality events that our system was configured to detect (see previous subsection). We instructed each grader to score a reported event as correct only if the following conditions are satisfied:

1. The representation correctly captures the semantics of the event reported in the headline.
2. The reported event is one of the 38 reality events that our system was configured to detect.
3. The reported event is not equivalent to another event.

Table 2 shows the result of this evaluation. Our system performed well across both graders, and the inter-grader agreement across all 10 technology trends was 0.8484 according to the Cohen kappa (Carletta 1996), which suggested substantial agreement.

We hypothesized that our system performed well for the following reasons.

1. Because our system’s NLU module can capture rich semantic representations of events from news headlines, the resulting representations are more accurate which results in higher precision.
2. Because our system’s flexible semantic matcher can detect equivalent events, our system can prevent over-counting which results in higher precision.

Hence, both components are necessary for good performance, and the ablation of either one will result in a significant drop in precision.

We tested these hypotheses by creating two variants of our system. We created a system called NoNLU by replacing our NLU module with a keyword-based approach. This approach matches keywords – associated with concepts in representations of reality events that our system was configured to detect – to news headlines to capture these representations. These keywords are provided by the lexical realizations of these concepts and their subclasses. We created

Tech Trend	N	Grader 1	Grader 2
BI	42	83.33	85.71
Cloud	95	84.21	91.58
IDM	12	83.33	83.33
Mashup	8	100.00	100.00
Mobile Platform	97	84.54	93.81
RIA	22	86.36	90.91
SaaS	72	84.72	86.11
SOA	32	84.38	87.50
Social Media	46	82.61	84.78
WiMax	42	90.48	95.24
Overall	468	85.04	89.96

Table 2: The precision of the reality events reported by our system over a three month period. The first column gives the technology trend. The second column gives the number of events reported. The third and fourth columns give the percentage of correct events according to graders 1 and 2 respectively.

Tech Trend	N	Grader 1	Grader 2
BI	50	68.00	72.00
Cloud	118	56.78	74.58
IDM	14	57.14	64.29
Mashup	10	80.00	80.00
Mobile Platform	127	59.84	69.29
RIA	29	58.62	65.52
SaaS	96	60.42	64.58
SOA	46	52.17	58.69
Social Media	82	47.56	54.88
WiMax	45	68.89	73.33
Overall	617	58.67	67.26

Table 3: The precision of the reality events reported by the NoNLU system. The second column gives the number of events reported by NoNLU.

another system called NoEqv that does not use a flexible semantic matcher to detect equivalent events.

We evaluated both systems on the same technology trends, using the same news sources, and over the same time period. The outputs of both systems were graded using the same methodology described above. Tables 3 and 4 show the results of this evaluation.

The overall precision of NoNLU was significantly lower than the overall precision of our system across both graders ($p < 0.01$ for the X^2 test in both cases and the Cohen kappa across all trends was 0.8384). Because NoNLU used only keywords to capture reality events from news headlines, many of the representations captured were incorrect (and hence lower precision). For example, NoNLU failed to correctly determine that Tatung is the recipient and Alcatel is the donor in the headline: *Tatung InfoComm purchases WiMAX base stations from Alcatel-Lucent*. Similarly, the overall precision of NoEqv was significantly lower than the overall precision of our system across both graders ($p < 0.01$).

Tech Trend	N	Grader 1	Grader 2
BI	90	41.11	41.11
Cloud	218	41.74	43.12
IDM	20	60.00	50.00
Mashup	15	53.33	53.33
Mobile Platform	219	45.66	50.68
RIA	33	60.61	63.64
SaaS	148	49.32	47.97
SOA	61	50.82	50.82
Social Media	77	57.14	59.74
WiMax	83	55.42	50.60
Overall	964	47.93	48.86

Table 4: The precision of the reality events reported by the NoEqv system. The second column gives the number of events reported by NoEqv.

for the X^2 test in both cases and the Cohen kappa across all trends was 0.8561). Because NoEqv does not check for equivalent events, it reported more than twice the number of reality events compared to our system. However, many of these events were equivalent because they were reported by different sources and in different ways. These results confirmed our hypotheses.

We observed, however, that the absolute number of correct reality events reported by our system was fewer than those reported by NoEqv for many of the technology trends. This difference suggested that our system incorrectly identified (and hence removed) events that were not really equivalent. We applied the X^2 test to these trends, but did not find any statistically significant difference. Hence, these differences can be attributed to chance.

Conclusion and Future Work

In this paper, we described the Technology Trend Tracker: a tool to measure the reality of technology trends of interest using information on the Web to inform decisions such as when to develop training, when to invest in expertise, and more. We gave an overview of the main features of this system followed by a description of the system's architecture.

We used our system to augment Accenture's annual technology vision survey and showed how our system achieved high precision on reporting reality events for the top ten technology trends from this survey. We also conducted an ablation study, which showed why our system performed well.

We are currently investigating the following issues before our system is ready for deployment to a wider community. We are evaluating the performance of our system on the metric of recall to understand what percentage of reality events our system is failing to report. We are exploring ways to enable end-users to author (and update) the technology trends and the reality events used by our system. This extension will make our system more accessible because a knowledge engineer will not be required to maintain these representations. Previous research on enabling subject matter experts to author semantic representations without the aid of knowl-

edge engineers (Barker et al. 2003) provides a promising solution. Finally, we are evaluating the utility of our system from an end-user perspective. Our system is currently deployed as part of a mobile application suite in a pilot program with a group of select users from Accenture. The use of our system in this context is enabling us to collect data on usage pattern and the utility of the information reported.

Acknowledgments

We want to thank the reviewers for helpful comments and suggestions. We also want to thank Lucian Hughes, Alex Kass, and Owen Richter for their contributions to this project.

References

- Baker, C.; Fillmore, C.; and Lowe, J. 1998. The Berkeley FrameNet Project. In *17th Intl. Conf. on Computational Linguistics*.
- Barker, K.; Blythe, J.; Borchardt, G.; Chaudhri, V.; Clark, P.; Cohen, P.; Fitzgerald, J.; Forbus, K.; Gil, Y.; Katz, B.; Kim, J.; King, G.; Mishra, S.; Morrison, C.; Murray, K.; Otsstott, C.; Porter, B.; Schrag, R.; Uribe, T.; Usher, J.; and Yeh, P. 2003. A Knowledge Acquisition Tool for Course of Action Analysis. In *IAAI*.
- Barker, K.; Porter, B.; and Clark, P. 2001. A Library of Generic Concepts for Composing Knowledge Bases. In *1st Intl. Conf. on Knowledge Capture*.
- Barker, K. 1998. *Semi-Automatic Recognition of Semantic Relationships in English Technical Texts*. Ph.D. Dissertation, University of Ottawa.
- Carletta, J. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. In *16th Intl. Conf. on Computational Linguistics*.
- Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
- Fillmore, C. 1971. Some Problems for Case Grammar. In *22nd Annual Round Table Meeting on Linguistics and Language Studies*.
- Lassila, O., and Swick, R. 1999. Resource Description Framework (RDF) Model and Syntax Specification. Technical Report W3C.
- Lenat, D., and Guha, R. 1990. *Building Large Knowledge-Based Systems*. Addison-Wesley.
- Sowa, J. F. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.
- Yeh, P., and Kass, A. 2008. Capturing the Semantics of On-line News Sources for Business Intelligence Applications. In *20th Intl. Conf. on Tools with Artificial Intelligence*.
- Yeh, P.; Porter, B.; and Barker, K. 2003. Using Transformations to Improve Semantic Matching. In *2nd Intl. Conf. on Knowledge Capture*.
- Yeh, P.; Porter, B.; and Barker, K. 2005. Matching Utterances to Rich Knowledge Structures to Acquire a Model of the Speaker's Goal. In *3rd Intl. Conf. on Knowledge Capture*.