Automated Identification of Terminological Dissonance in IT and adjacent fields

Ms. Jessica Richards, BYU

Graduate student in Information Technology with a background of interdisciplinary work between computing and media fields. Highly interested in streamlining the collaborating of technical and creative minds.

Joseph J Ekstrom, Brigham Young University

Dr. Ekstrom spent more than 30 years in industry as a software developer, technical manager, and entrepreneur. In 2001 he helped initiate the IT program at BYU. He was the Program Chair of the Information Technology program from 2007-2013. His research interests include network and systems management, distributed computing, system modeling and architecture, system development, Cyber security and IT curriculum development.
Automated Identification of 
Terminological Dissonance in IT and 
adjacent fields

ABSTRACT
Information Technology often fills the role of tool supplier to other disciplines. This role necessitates that IT academics and professionals perform constant interdisciplinary communication. Semantic “dissonance”, in the forms of synonymy and polysemy, is frequently encountered between participants in related meetings and discussions.

Unfortunately the topic of semantic miscommunication is usually not broached until it causes a project meltdown. This laissez-faire approach can be compared to an information security manager ignoring potential virus threats until a machine is already infected.

Taking a more practical stance towards the problem, we developed the Termediator software to pre-emptively identify potential term dissonance. Termediator has evolved since 2010 from a simple term browser to a multifaceted tool; in its current state it integrates similarity measures in synonymy with topic modeling and clustering in polysemy.

The professional uses of Termediator include collaborative projects (both inter- and intradisciplinary) and telecommuting work situations. Termediator also has a distinct role in IT education, where it is imperative to include pedagogy that sensitizes students to the potential for misunderstanding because of semantic differences in commonly used terms.

1. INTRODUCTION
Cognitive dissonance refers to a situation when an individual is simultaneously holding two contradictory beliefs. The term was coined in 1954 by psychologist Leon Festinger, who proposed the combined presence of contradictory beliefs produces psychological discomfort in the individual, and the greater the discomfort, the greater the desire to reduce the dissonance of the two cognitive elements.

This definition of cognitive dissonance limits itself to an individual mental experience. But what if the dissonance is not in the chattering of one’s own brain, but between two people in a conversation? Consider a humorous example of this situation:

“When told to ‘secure’ a building it has been related that,
• The Navy issues a purchase order for the building.
• The Air Force locks the doors and turns on the alarm system.
• The Army evacuates the personnel, then locks the doors and turns on the alarm system.
• The Marines assault the building using ground troops and air support, and then deploy squads in and around the building checking the credentials of all who aspire to enter the building.

In this example, the word “secure” was attached to different meanings. Technically speaking, the term “secure” is a “polysemous” term. Polysemy means “many signs” and is used when one sign
is linked to different definitions. In a linguistic setting a “sign” refers to a written term, which is a word or a phrase.

It is helpful to see how a polysemous term plays out abstractly in this visualization of a conversation:

![Figure 1. Polysemy illustrated in conversation.](image)

Each person is using the same “term” (referred to by the scribble marks in figure) in a conversation, yet each person associates that term with a different concept (indicated by the colored shapes in the figure). Since the term is the sole representation of the concept, the message syntax is received correctly but interpreted incorrectly.

Some have referred to this phenomenon as simply “miscommunication,” however the problem extends beyond the meaning of that term; instead we use the word “dissonance.” Miscommunication typically refers to one meaning that was not adequately communicated (e.g. I said “mark” and you heard “shark”). Instead, we use dissonance to refer to two or more valid yet conflicting meanings that were conveyed in the same conversation.

Dissonance is especially common when the conflicting meanings are similar but not identical. As indicated in the figure, each person was referring to a blue object, but the shape of the blue object was different for each individual. These people may talk for a long time about blue objects, thinking that they are discussing the same blue object, before they realize that one means a blue star-shaped object, one means a blue triangle-shaped object, and one means a blue diamond-shaped object.

An IT professional named “Bob” is instructed to develop software for the military. The software is intended to support the action of “securing” headquarters.

- For the Navy, Bob needs to write financial support software that would enable them to issue a purchase order for the building
- For the Air Force, Bob creates software that automatically locks the doors and switches on the alarm system.
- For the Army, Bob needs to develop a program that facilitates personnel evacuation and alarm activation.
- For the Marines, Bob needs to create a deployment program that will support a full-scale assault on the building.

One may say that this example is a bit hyperbolic, so let’s look specifically at a more technical term that has several specific meanings that are still radically different. The term “ATM” occurs in four communities that frequently interact: finance, technology, biology, and medicine.
### Table 1. Definitions of polysemous term \textit{ATM}.

| \textit{ATM} in \textbf{finance} | \begin{itemize} \item A computerized electronic machine that performs basic banking functions (as handling check deposits or issuing cash withdrawals). Also called automated teller machine\textsuperscript{30}. \end{itemize} |
| \textit{ATM} in \textbf{technology} | \begin{itemize} \item The ITU standard for a cell-relay based communications system encompassing voice, data and video traffic. ATM provides standards for 25Mbps and 155Mbps transmission speeds. Because of the expense of the architecture, most networks do not handle this all the way to the workstation but larger networks will use it as a backbone. The unique function of this over other backbones other than speed is the self handled ability to prioritize traffic and requests. \textsuperscript{9} \end{itemize} |
| \textit{ATM} in \textbf{biology} (first definition) | \begin{itemize} \item Ataxia telangiectasia mutated. A checkpoint kinase which transduces genomic stress signals to stop cell cycle progression and promote DNA repair, acting via p53, a tumour suppressor protein. Its cognate gene, ATM (see below), is mutated in ataxia telangiectasia, a rare neurodegenerative disease characterised by ataxia telangiectasias, increased chromosome fragility when exposed to ionising radiation and predisposition to lymphomas\textsuperscript{41}. \end{itemize} |
| \textit{ATM} in \textbf{biology} (second definition) | \begin{itemize} \item A gene on chromosome 11q22-q23, which encodes a PI3/PI4 cell-cycle checkpoint kinase that phosphorylates, thereby regulating a broad range of downstream proteins—e.g., tumor suppressor proteins p53 and BRCA1, checkpoint kinase CHK2, checkpoint proteins RAD17 and RAD9, and DNA repair protein NBS1\textsuperscript{41}. \end{itemize} |
| \textit{ATM} in \textbf{medicine} | \begin{itemize} \item Atmosphere, atmospheric\textsuperscript{31}. \end{itemize} |

Now, perhaps the financier will never have a conversation with the biologist that brings the conflicting definitions of ATM to light. However, an IT professional could easily encounter
facets of medicine, biology, and finance just by contracting with one company. It is not entirely implausible that an IT professional may be required to set up an automated teller machine in a building that uses an asynchronous transfer mode network to communicate with others about their work on ataxia telangiectasia mutated.

These examples illustrate a subtle communications problem. When one hears unknown words, such as in a foreign language, the failure to communicate is obvious. However, when one hears words that sound correct in the context, the failure to communicate is not realized and sometimes produces serious consequences. There is humor in miscommunication; however, it is not funny when a project fails because of a misunderstanding—especially when it could have been prevented.

2. BACKGROUND
This research effort was motivated by the observation and experience of communication difficulties during IT system development. The most difficult part of defining requirements is coming up with a common model and vocabulary to describe the domain and function of the new system. The problem also appears when attempting to rationalize vocabulary between IT and its adjacent disciplines. Information Technology is a computing discipline that shares a heritage with Computer Science, Software Engineering, Computer Engineering, and Information Systems. Other related domains include Business Process Management, and Systems Engineering—just to name a few. Being the tool supplier for almost everyone necessitates that IT academics and professionals perform constant interdisciplinary communication. Semantic “dissonance” is frequently encountered between participants in meetings and discussions. When two collaborators use the same word to mean different things, even a slight definitional difference can create a serious roadblock that not only frustrates the collaborators but impedes the progress of any joint project.

2.1 Synonymy
Recall that of the two types of dissonance, there is synonymy and polysemy, “Synonymy” is when two distinct words or phrases have the same or similar meanings.

![Figure 2. Diagram of synonymy.](image)

Examples of common synonyms include:
- Buy and purchase
- Big and large
- Quickly and speedily

Synonymy may not garner too much difficulty in everyday speech—most people know that “big” and “large” can be used interchangeably. But when the terminology is more specialized and technical, synonymy can prove frustrating. For example:

*byte*: The number of bits used to represent a character. For personal computers a byte is usually 8 bits.

*character*: A single letter, gure, punctuation mark, or symbol produced by a keystroke on a computer. Each character is represented by a byte.
Depending on the person’s background, it may not be immediately apparent that *character* and *byte* are often used to refer to the same concept. It is helpful to know synonymous or near-synonymous pairs such as these in collaborative conversations.

### 2.2 Polysemy

“Polysemy” is the potential for a term to have multiple meanings.

**Figure 2.** Diagram of polysemy.

One common example of polysemy is the word “crane”:

- a bird
- a type of construction equipment
- to strain out one’s neck

A more technical example is the word “process”, which is used constantly across multiple domains.

*process*: A set of interrelated activities which transform inputs into outputs.

*process*: An executable unit managed by an operating system scheduler.

As you can see, depending on the field and the context, the meaning of *process* can vary both in definition and specificity. This is the essence of polysemy.

### 3. DISSONANCE

Many of us have participated in discussions that were resolved only after all of the participants agreed to a common vocabulary. That is, we had to agree to a “glossary of terms” in order to communicate. Teams always seem to develop a set of acronyms and terms specific to the team. But what happens when multiple teams from different fields are working together, each with our own team-specific terminology? Is there a way to know ahead of time where miscommunication is likely in order to accelerate the vocabulary normalization phase of teambuilding? Can a synthesis of existing technical glossaries be analyzed to create “warning lists” of dissonant terms?

This glossary-centered approach has been previously attempted at the intersection of Systems and Software Engineering. The result was the ISO/IEC 24765 and the *sevocab*, an aggregated glossary and its associated website. However, an aggregation of only two disciplines is not broad enough to satisfy the needs of IT. For example, the term “enterprise architecture” had 0 hits in the *sevocab*. The *sevocab* interface does not allow for general browsing of term relationships that would allow the user to manually identify synonymous or polysemous terms. However, interesting data definitely existed in sevocab: the term “system” had 8 concept descriptions, and there were several conflicts in term usage apparent. We hypothesized that similar data could be used with a different interface to discover dissonance in terminology.

It should be noted that language ambiguity detection is not a new area of research. Term ambiguity detection (TAD) frameworks have been developed\(^7\) that attempt to identify ambiguity from a general English language corpus. In addition, lexical databases such as Dante provide a record of relationships between words that provides insight into ambiguity in general language\(^27\). Such lexicons often pair with NLP applications to provide information extraction (IE) or ambiguity detection systems, however their scope includes all terms in a language (or multiple
languages) and is therefore exceedingly broad. For example, the Corpus of Contemporary American English\textsuperscript{13}, or COCA, contains 400 million words pulled from magazines, newspapers, spoken recordings, fiction, and academic journals\textsuperscript{14}. Likewise, the Dante project uses a compilation of corpuses, including the British National Corpus and the Hiberno-English Corpus, totaling over 1.7 billion words of general English\textsuperscript{12}.

Our research therefore differs from our comrades in linguistics primarily by two factors: corpus and objective. We are not attempting to generally identify ambiguity from billions of words in general English. Our research only uses data from \textit{controlled vocabularies} within Information Technology and related specialized subject fields (SSFs). These controlled vocabularies are authored by communities of experts, lack structured prose, and mainly consist of very short text definitions composed of sentence fragments. Narrowing this scope to not only languages for special purposes (LSPs), but also to controlled vocabularies using LSPs, gives us insight into dissonance between experts in intra- or inter-disciplinary communication\textsuperscript{7}. With this specific interest in dissonance specifically within specialized terminologies, and the semantic distance between terms\textsuperscript{39}, we began our first glossary aggregation prototype.

\textbf{3.1 2010 Glossary Aggregation Prototype}

In 2010 the first prototype of the Termediator tool was created. This was the first attempt to build software to investigate and attempt a partial solution for synonymy and polysemy. This prototype parsed and normalized the ISO/IEC 24765 (sevocab) data into Python `dict` data structures. To grant web access to the data, we used a Django (Django Project, 2013) interface paired with the dictionary persisted in SQLite as the database. Through this interface we sought to create a way to explore the terminology in ways that the sevocab did not allow.

The main function implemented was a web “term browser” that allowed the user to browse terms by how many concepts they had. Sorting high to low on the number of associated concepts is useful when searching for potentially dissonant terms.

Although the research at this time\textsuperscript{17} was very preliminary, the work performed on this initial prototype gave us the framework for more sophisticated tools in the years to come. The current revision of the tool can be found online at http://termediator.byu.edu:8080

\textbf{3.2 Termediator and Synonymy Identification}

What followed the initial glossary aggregation prototype was the “Termediator” tool: this tool’s end goal was to automatically identify synonymous dissonant terms between two or more fields\textsuperscript{38}. Recall that of the two types of dissonance, there is synonymy and polysemy, and at this point the tool only focused on detecting synonymy. There was a lot of work to be done to reach that point, and the first step was to create a standardized method for data input and normalization. We created an XML 1.0 schema, and then changed all of our existing parsers to output to that XML within that schema structure. A merging program was developed that combined all of the XML outputs into a compendium of glossary data. With a standardized input, output, and merging chain in place, we proceeded to quadruple the size of our data set and broaden its reach by bringing in glossaries from over fifteen overlapping domains of interest.

The first problem Termediator tackled was detecting synonymy, or when term A and term B share Concept C. To identify synonymous terms, a vector model “similarities matrix” was created to compare every concept with every other concept; each relationship was then assigned a similarity ranking. A perfect similarity ranking of 1 meant the concepts were identical, and anything close to 1 meant the concepts were very similar. Termediator then linked each concepts to its 3 most similar concepts in the web interface. At this point, there was not yet an automated
way to list synonymous terms, they could only be identified by manually browsing through Termediator’s term list.

### 3.3 Polysemous Semantic Clustering

The next step for the Termediator tool was to attempt to identify polysemy, or when a word or phrase is linked to multiple conflicting concepts. Consider that the intuitive way for a human to find a polysemous term is to look at a term’s concepts and sort them into groups by meaning. If there are many groups of meaning, then it may be reasonable to assume that the term is polysemous. Recall the term “ATM” from *Table 1.1*. This term has four meaning groups: finance, networking, biology, and medicine. Within these groups we listed five distinct definitions in *Table 1.1*. Clearly these concepts listed under the same term have dramatically different meanings. With this observation in mind, the hypothesis was made that a term would be polysemous if it contained a high number of semantic concept clusters, or “groups of distinct meaning.” If Termediator could automatically sort concepts into these semantic groups, then we could see which terms had the most clusters and therefore the most potential for dissonance.

#### 3.3.1 Automated Clustering

Termediator’s clustering process used the hierarchical agglomerative algorithm because it did not require a predetermined number of clusters. Predetermining the number of concept clusters would be a difficult manual process, since the number of true semantic clusters would vary term to term. Hierarchical agglomerative clustering places each concept into its own cluster initially, and then systematically combines concepts into groups using an associated proximity matrix.

To build the proximity matrices for the clustering method, text similarity measures were needed that would indicate how similar one concept is to another. To produce these values, three different similarity algorithms were used: cosine, latent semantic indexing (LSI), and latent Dirichlet allocation (LDA). Cosine is a simple vector measure that transforms each text concept into a numerical vector. The similarity between two concepts is determined by taking the cosine of the angle between the two vectors. LSI assumes that words used in similar contexts have similar meaning. Using singular value decomposition, LSI identifies similarities between texts even if they don’t share similar wording. Lastly, LDA identifies distributions of words (also known as topics) within a particular corpus. Concept similarity is based on the degree to which they share those topics. As both LSI and LDA require a training corpus, the glossary compendium was used as the training corpus. All three of these similarity methods produce concept similarity values between zero and one (higher values indicate more similarity between two concepts). Using these values, Termediator1.5 then generated the proximity matrix for each term.

With the proximity matrices in hand, linkage types were added as the next clustering parameter. The linkage type is necessary to measure similarity between clusters of concepts (which is more difficult than comparing just one concept to one other concept). We initially looked at three linkage types: single, complete, and average. We chose not to evaluate single linkage because prior research has proven that it “generally gives results that are far inferior to those obtainable when the other hierarchical agglomerative methods are used” [reference]. We then evaluated average and complete linkage and determined that both should be included as options in our clustering method.
3.3.2 Determining Threshold

To make the clustering data useful, each term needs a distance to determine what concepts are clustered together, a clustering “threshold”. Intuitively, the threshold is where the term’s relationship dendrogram is sliced. A dendrogram is a tree diagram used to illustrate cluster arrangement in hierarchical clustering. At the best threshold, all the concepts in a term’s tree diagram are categorized in the appropriate semantically related groups. To find this candidate threshold, we initially attempted crowdsourcing the data through a web application coupled with a survey.

![Figure 3. Cluster headings in the web application.](image)

The interface shown above allowed users to adjust a slider that changed the threshold and adjusted the dendrogram inspired cluster visualization accordingly. Even at this high threshold, the term “process” has a high potential for dissonance as it contained 9 semantic groups.

![Figure 4. Definitions under one cluster in the web application.](image)

Each group contained all the concepts deemed to be most similar to each other. Shown above is “group 7” from the term “process.” Users were expected to experiment with the application’s slider, and then select the approximate threshold that produced the most accurate number of groups. If a user saw a concept within a group that was “not like the others” then they needed to change the slider until more similar groups were formed. However, the complexity of analyzing multiple definitions in fluid clusters proved too difficult for most users, as less than 10% of users enticed to the site actually provided any data.

Although crowdsourcing did not produce the needed data, the application’s visualization features provided insight into a potential measure of interest. Every term has a threshold at which it converges all text concepts into a single cluster. Terms that we manually identified as “simple” converged at lower threshold values than “complex” terms. This threshold can best be understood as the distance between the clusters in the hyperspace defined by the analysis method. Higher thresholds mean that points that are farther apart are grouped in the same cluster, which led us to an application for this threshold in polysemy identification. Discovering this threshold and its application was a key step in developing the polysemy portion of the Termediator tool.

Guided by the insight that the threshold where all text concepts were close enough to be collapsed into a single cluster was a useful measure of potential polysemy, we ran all three clustering methods on every term and recorded the threshold when all of the concepts converged into one group; we called these threshold values “convergence values.” Creating a simultaneous
graph of every convergence value of every term illustrated a trend that persisted regardless of clustering method or linkage type. Combining the convergence values with the mean we were able to generate a candidate threshold value for each clustering and linkage combination. Graphing all of these candidate thresholds at the same time (see figure below) revealed the trend persists for all similarity measures and linkage types.

Figure 5. Cluster convergences across the entire dataset.

The matrix of candidate thresholds was then utilized to perform hierarchical clustering on each term in the compendium and sorted the results by cluster frequency. The result is a table of terms with the most clusters for each of the six clustering-linkage algorithms: LSI complete, LSI average, LDA complete, LDA average, cosine complete, and cosine average. Consider the table for LSI average:

<table>
<thead>
<tr>
<th>Top Clustered Terms (LSI average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. interface (15 clusters)</td>
</tr>
<tr>
<td>2. function (11 clusters)</td>
</tr>
<tr>
<td>3. object (10 clusters)</td>
</tr>
<tr>
<td>4. unit (9 clusters)</td>
</tr>
<tr>
<td>5. standard (9 clusters)</td>
</tr>
<tr>
<td>6. process (9 clusters)</td>
</tr>
<tr>
<td>7. node (9 clusters)</td>
</tr>
<tr>
<td>8. link (9 clusters)</td>
</tr>
<tr>
<td>9. firewall (9 clusters)</td>
</tr>
<tr>
<td>10. firewall (9 clusters)</td>
</tr>
</tbody>
</table>

The results in this table are interesting because they include many terms that can be manually identified as polysemous terms. Words ranked highly in these results, such as “function,” “process,” and “resource,” are fraught with potential for miscommunication. One may be concerned that this is a simply a list of terms that have a high number of concepts, but that is not the case. Terms with even higher numbers of concepts, but a low number of semantic groupings, are generally weeded out by this clustering strategy. Thus this convergence threshold, or where a term’s concepts are clustered into the most semantically relevant groups, provides an automated method for identifying potentially polysemous terms from domain specific glossaries, in spite of the accuracy problems associated with analysis of short texts such as glossary concepts.

If we can find a convergence point by running a cluster algorithm, and combine the matrix of all convergence values with the mean to find a candidate threshold, can we set the threshold value beforehand? The answer is both yes and no. The candidate threshold value is corpus and algorithm dependent, meaning that the value varies depending on which algorithm is used and
the exact status of the corpus at the time. The candidate threshold is different for each clustering and linkage combination—which gives us a total of nine distinct candidate thresholds. Additionally, every time a definition is added to the corpus, the end convergence point for any one term may change. If the convergence points for terms change, this affects the overall dendrogram and thus changes the "candidate threshold" for any clustering-linkage type. Although we can see that the trend persists when graphing each of the nine candidate thresholds, that observation does not produce an absolute threshold value that generates identical results regardless of algorithm or corpus.

4. POLYSEMY IDENTIFICATION
The focus of this part of Termediator was to combine our database of terms and concepts with the convergence values from the automated clustering feature, as well as the interface from the user survey web application. This combination provided the means to create a “polysemy” tool that ran alongside the original “synonymy” tool. The polysemy tool identifies terms that have a high number of semantic clusters, which means the term is potentially polysemous. Terms that are potentially polysemous are candidates for confusion in communication. The polysemous facet of Termediator identifies potentially dissonant terms within one domain, across all domains, or in a subset of domains in the database.

4.1 Polysemy Tool Overview
The polysemy tool interface looks a lot like the clustering survey application used previously: it shows a term and its concepts in semantic clusters that can be changed via a threshold slider. In addition, the polysemy tool adds utilization of all terms and concepts in the database, and also provides the ability to filter results by domain(s). The figure below shows the new tool division on the Termediator web app and the view of a single term in the polysemy tool.

Figure 5. Polysemy web tool interface.

The polysemy tool allows you to search for a single term, or automatically generate the most dissonant terms in a set of domains.

Findings from the polysemy tool are described in the results section.

5. RESULTS
As semantic clustering was only performed in a polysemous context, the results focus on Termediator’s ability to identify polysemous and potentially dissonant terms. An analysis of
Termediator’s synonymy results was previously published\textsuperscript{38} and used as a foundation for this current work.

The polysemy portion of Termediator has two main features. The first is the ability to search for a single term and view its concepts in semantic clusters whose threshold can be adjusted by the user. The second feature is the ability to generate lists of highly dissonant terms within a domain or set of domains. However term browsing of all terms in the database (without a specific query) is not yet implemented in the polysemy tool.

\textbf{Figure 7.} Searching for a polysemous term.

\textbf{Figure 8.} Viewing and adjusting a term’s concepts within semantic clusters

The synonymy portion of Termediator does not yet implement semantic clustering or dissonant term list generation. The synonymy tool does allow general browsing of all the terms in the database, as well as their concepts and top synonymy matches for those individual concepts. The results can also be filtered to find a specific term and to only lists concepts sourced from a particular domain. Future work should focus on applying semantic clustering and dissonant term list generation to the synonymy half of Termediator.

\textit{Browsing synonymous terms with filtering options}
5.1 Polysemy Results

The first results of the polysemy tool are the top dissonant terms (terms with the most semantic clusters) within individual domains. Words that appear in a domain’s top dissonance list have the highest potential for conflict, among practitioners in that domain, as best discerned by our current tool.

Currently there is not a similarity measure and linkage type that is definitively the “best” for all results in all domains. One measure-linkage combination may produce better results for Information Security, and another is a stronger fit for Requirements Engineering. Which results are “best” is also open to argument due to the subjective nature of linguistic interpretation. The findings are constrained to the seven domains (out of fifteen) that have the most terms and concepts in the database. The accuracy of the results increases dramatically with a larger data set. The best semantic grouping out of the six similarity measure-linkage type combinations was chosen manually for each domain. From that grouping, the top term and a sampling of concepts underneath that domain are given to illustrate the term’s internal dissonance.

5.1.1 Information Technology

<table>
<thead>
<tr>
<th>LSI average</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. standard</td>
<td>2. firewall</td>
</tr>
<tr>
<td>3. access</td>
<td>4. redundancy</td>
</tr>
<tr>
<td>5. post</td>
<td>6. interface</td>
</tr>
<tr>
<td>7. interactive</td>
<td>8. hierarchy</td>
</tr>
<tr>
<td>9. error</td>
<td>10. data</td>
</tr>
</tbody>
</table>

“standard”
• An approved model\textsuperscript{15}.
• A widely accepted way of doing something\textsuperscript{5}.
• A mandatory technology, result or procedure to be applied in all appropriate situations\textsuperscript{32}.

5.1.2 Information Security

<table>
<thead>
<tr>
<th>LDA average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. spoofing</td>
</tr>
<tr>
<td>2. worm</td>
</tr>
<tr>
<td>3. spam</td>
</tr>
<tr>
<td>4. risk</td>
</tr>
<tr>
<td>5. firewall</td>
</tr>
<tr>
<td>6. virus</td>
</tr>
<tr>
<td>7. race condition</td>
</tr>
<tr>
<td>8. payload</td>
</tr>
<tr>
<td>9. zombie</td>
</tr>
<tr>
<td>10. whitehat</td>
</tr>
</tbody>
</table>

“spoofing”
• Impersonating another person or computer, usually by providing a false email name, URL, domain name server, or IP address\textsuperscript{26}.
• A generic label for activities in which trusted relationships or protocols are exploited for mischievous or surreptitious ends especially those cases in which an unknown or unauthorized actor surreptitiously pretends to be a trusted one. The spoofing need not entail personal identification tactics in which a machine's identity or address data are usurped are also termed spoofing\textsuperscript{18}.
• Spoofing means a router responds to a local host in lieu of sending information across a WAN link to a remote host. The local host thinks the response came from the remote host/network, when it really came from the router\textsuperscript{45}.

5.1.3 Graphic Design

<table>
<thead>
<tr>
<th>Cosine average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. dummy</td>
</tr>
<tr>
<td>2. pixel</td>
</tr>
<tr>
<td>3. margin</td>
</tr>
<tr>
<td>4. font</td>
</tr>
<tr>
<td>5. typography</td>
</tr>
<tr>
<td>6. typeface</td>
</tr>
<tr>
<td>7. thumbnail</td>
</tr>
<tr>
<td>8. template</td>
</tr>
<tr>
<td>9. resolution</td>
</tr>
<tr>
<td>10. register mark</td>
</tr>
</tbody>
</table>

“dummy”
• A rough form of any document\textsuperscript{1}.
• A small, detailed page diagram showing where all elements go\textsuperscript{21}.
• A dummy counts as an example of a piece of design work (brochure, ad, book cover etc.)
that needs to be approved by the client. Once the client approves the dummy, the designer creates and prints the final design.

5.1.4 Software Engineering

<table>
<thead>
<tr>
<th>LDA complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. abstraction</td>
</tr>
<tr>
<td>3. version</td>
</tr>
<tr>
<td>5. object</td>
</tr>
<tr>
<td>7. design</td>
</tr>
<tr>
<td>9. complexity</td>
</tr>
</tbody>
</table>

“abstraction”
- Generalization, ignoring or hiding details. Examples are abstract data types (the representation details are hidden), abstract syntax (the details of the concrete syntax are ignored), abstract interpretation (details are ignored to analyse specific properties).
- Parameterization, making something a function of something else. Examples are lambda abstractions (making a term into a function of some variable), higher-order functions (parameters are functions), bracket abstraction (making a term into a function of a variable).
- A cohesive model of data or an algorithmic procedure.

5.1.5 System Engineering

<table>
<thead>
<tr>
<th>LSI complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. constraint</td>
</tr>
<tr>
<td>3. task</td>
</tr>
<tr>
<td>5. process</td>
</tr>
<tr>
<td>7. object</td>
</tr>
<tr>
<td>9. implementation</td>
</tr>
</tbody>
</table>

“constraint”
- Restriction on the value of an attribute or the existence of any object based on the value or existence of one or more others.
- Restriction on software life cycle process (SLCP) development.
- Limitation or implied requirement that constrains the design solution or implementation of the systems engineering process and is not changeable by the enterprise.

5.1.6 Business Process Management

<table>
<thead>
<tr>
<th>LDA average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. process</td>
</tr>
</tbody>
</table>
“process”

- A set of interrelated activities, which transform inputs into outputs.
- The step-by-step sequence of activities (systematic approach) that must be carried out to complete a project.
- An activity that is part of a data flow diagram. Systems can be built to process goods or to process data. Most information system work focuses on processes that alter data.
- an executable unit managed by an operating system scheduler.

5.1.7 Workflow

```
<table>
<thead>
<tr>
<th>LDA complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. task</td>
</tr>
<tr>
<td>3. baseline</td>
</tr>
<tr>
<td>5. transparency</td>
</tr>
<tr>
<td>7. spike</td>
</tr>
<tr>
<td>9. link</td>
</tr>
</tbody>
</table>
```

“task”

- required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a process.
- activities required to achieve a goal.
- a process that cannot be subdivided any further: an atomic process.
- The goals, steps and skills needed to accomplish an activity; a task may comprise a series of sub tasks for instance the task of making tea may involve the sub task of filling the kettle.

5.1.8 Other Domains

While Termediator has a wealth of entries in its glossary compendium, the majority of those glossaries originate from certain domains. Termediator has not yet acquired enough data within the following domains to support single-domain analysis of the clustering results: Computer Science, Electrical Engineering, Enterprise Architecture, Information Systems, Requirements Engineering, Robotics, Telecommunications, and User Experience Design.
5.2 Top Multidisciplinary Dissonant Terms

One of the main purposes of Termediator is to identify terms that are potentially polysemous and therefore potentially dissonant in multidisciplinary projects. Described below are three sample teams that represent common groupings of disciplines in collaborative projects. A top term is then chosen from Termediator’s list of potentially dissonant terms across the selected domains. Each discipline’s definition is listed in order to see the discrepancy between each team member’s understanding of the term.

5.2.1 IT + SSE + IS + SE

The top dissonant term in this team using “cosine complete” similarity-linkage is: “risk”

- **IT**: A condition or action that may adversely affect the outcome of a planned activity.
- **SSE**: Combination of the probability of an event and its consequence.
- **IS**: The level of impact on organizational operations (including mission, functions, image, or reputation), organizational assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring.
- **SE**: Risk is defined as the possibility of loss or injury. Risk exposure is defined by the relationships.

5.2.2 IT + CS + Graphic Design

Creating a website often involves collaborative teams of information technologists, computer scientists, and graphic designers. Typically the information technologist sets up and maintains the server, the computer scientist develops the website backend, and the graphic designer creates the frontend.

The top dissonant term in this team using “lsi complete” similarity-linkage is: “interface”

- **IT**: A point of connection or junction.
- **CS**: Class definitions and method signatures provide interfaces. Application program interfaces (APIs) form the interface of a system to applications and often consist of collections of functions or commands in a scripting language. Interfaces may be hidden (available only to the system developer) or exposed (available to others).
- **GD**: The front-end is basically the opposite of the back-end. It’s all the components of a website that a visitor to the site can see (pages, images, content, etc.) Specifically, it’s the interface that visitors use to access the site’s content. It’s also sometimes referred to as the user interface.

5.2.3 BPM + WF + ISYS + IT

Creating a functional workflow management system for a big corporation requires the collaboration of workflow analysts and business process managers to create and maintain the flow, as well as information technologists and information systems experts to coordinate the system backend and integration into the company network.

The top dissonant term in this team using “cosine average” similarity-linkage is: “data”

- **BPM**: Defines the type of information exchanged between business processes.
- **WF**: Data elements can be defined by tasks that are accessible only within the context of...
individual execution of that task\textsuperscript{46}.  
• **ISYS:** Consists of factual elements (or opinions or comments) that describe some object or event. Data can be thought of as raw numbers or text\textsuperscript{34}.  
• **IT:** Computer data is information processed or stored by a computer. This information may be in the form of text documents, images, audio clips, software programs, or other types of data\textsuperscript{31}.  

5.3 Practical Use of Termediator
Termediator is more than an exploratory tool. Even at this early stage in development, it offers practical applications for both professionals and educators.

5.3.1 Professional Collaborative Projects
Consider the project manager who is managing a multidisciplinary team. Preventing miscommunication is key to productive output and high morale when team members hail from different disciplines. The project manager can take note of confusing terms used by team members and check them in Termediator for conflicts. For synonymy, Termediator might help the team realize that Michael uses “character” and Samantha uses “byte” to mean the same thing. For polysemy, Termediator could indicate that Gerald and Hannah are not specifying which “interface” or what “system” in their discourse.

Project managers can also prepare their teams at the outset by creating “warning lists” with Termediator’s polysemous term list generation tool. The project manager would select the domains included in the team and generate a list of top potentially dissonant terms. The first team meeting could include a rundown of potentially dissonant terms and definitions from each team member’s domain.

5.3.2 Telecommuting
Awareness of potentially dissonant terms is especially relevant in this day and age of telecommuting and global outsourcing. More communication is being performed via text and email rather than in-person or on the phone; even with the advent of video conferencing, it is often more convenient to shoot an email than to Skype with someone five time zones away. For projects where textual communication is dominant, semantic understandings often go unnoticed until a major project flaw occurs. Rather than wait for a train wreck, professionals can and should use Termediator to identify potentially dissonant terms in their collaborative emails.

5.3.3 Information Technology Education
One of the original intentions of Termediator was to benefit students and educators in Information Technology. Because Information Technology sits adjacent to many other disciplines, it is imperative that IT education includes pedagogy that sensitizes students to the potential for misunderstanding because of semantic differences in commonly used terms.

While some more isolated fields still operate under the mindset that “their” definition of a term is canon, someone in IT will work with other fields their entire career and therefore they must recognize the semantic shades of gray. It must also be recognized that when semantic dissonance is encountered frequently, it is not enough to “roll with the punches.” Would you tell an Information Security analyst to ignore potential virus threats until one actually infects a machine? Of course not! Clear communication is absolutely essential for the success of IT projects; this is the professional reality that IT students must be prepared to face after graduation. Just as we teach students to prepare for malware or system failure, we should also teach students to prep for effective collaboration and communication with adjacent disciplines. The real
problem of miscommunication must be personalized so the student recognizes that “this will be an issue for me in my actual career.”

As a tool suite, Termediator can be used to sensitize students to the semantic misunderstandings that will occur in their professional careers. For example, when a student looks at a top ten list of polysemous terms as identified by Termediator, they may immediately recognize the many meanings of "system" and "interface". These meanings vary within IT alone, and by selecting concept comparisons between related fields, the student can see how these words may be misinterpreted when they collaborate with a computer scientist or a graphic designer. If the complexity of a term is not as obvious, they can view the clustering interface to see all of the definitions under a polysemous term. The included author and discipline source of the definition gives additional insight into the varying contexts of a term's usage. The synonymy portion of the tool not only functions as a multi-discipline database of terminology, but as a synonymy identifier. By drilling down into a definition, a student can use Termediator to visualize how similar meanings are shared between different terms. As in the polysemy section, the author and discipline information is readily available, which makes terminological, interdisciplinary sensitization through the tool more effective. Overall, Termediator is a tool that proposes confusing terms in an automated way in order to assist human agents in identifying problematic areas in intra- and -interdisciplinary communication. This can generate paradigm shifts in students that are accustomed to communicating primarily with likeminded peers and professors.

In addition to sensitizing students to the grey areas of their own terminology, Termediator can also be used to troubleshoot communications that occur in multidisciplinary academic collaborations, as many educational programs already integrate IT with adjacent disciplines. Termediator can also help transition faculty from other programs (e.g. CS or ISYS transfers), create an awareness of synonymy and polysemy in intro level classes, and produce more productive panels in multidiscipline conferences.

6. CONCLUSIONS

Much of our research can be described as topic analysis using short text concepts. While most topic analysis uses larger pieces of text from huge corpuses of data, there has been work done with short text concepts to inspect Twitter and other microblogging data. Microsoft’s Twahpic is one tool that uses PLDA (partially labeled latent dirichlet analysis) to identify 200 general topics (e.g. sports, politics, fashion, etc.) present in current tweets.

We also must mention the wealth of topic analysis occurring in the linguistics field. Techniques in natural language processing such as WSD (worse sense disambiguation) are being applied to multilingual problems, such as translating ambiguous words to their equivalents in another language. However, the data set for most multilingual problems is somewhat endless because there is so much “general” text available for analysis.

A major contribution of Termediator and its related research is how it applies topic analysis on short entities of “technical” (rather than “general”) text, and then make semantic connections between domains of study. But the truly unique contribution is how this research uses topic analysis to mitigate a specific communication problem.

The current revision of the program is capable of identifying terms that are candidates for dissonance in either the synonymous or polysemous realm, although the feature set for these realms is not currently identical and is more developed for the polysemy tool.
For synonymy, a user can search for a problematic term and see its top matches for dissonance (other terms that have the same meaning). All the terms in the dataset are available for alphabetical browsing in the synonymy tool, and they can be sorted be filtered by domain, similarity measure, concept count, and term length.

For polysemy, a user can also search for a term and view all of its concepts within semantic clusters to view the many meanings of a polysemous term. The polysemy tool also allows deeper analysis of a term’s semantic groups; a user can change the similarity measure and linkage type, and also adjust the algorithm threshold to view how similar semantic clusters are to another. Lastly, the polysemy tool also has a “warning list” generation feature, where the user can a select a domain or set of domains and generate a list of terms with the most potential for polysemous dissonance (terms with the most semantic clusters of concepts). This ability to generate warning lists is particularly helpful for practical use in the field.

While topic analysis applications have been previously developed, Termediator is first program we know of that applies vector similarity measure and semantic clustering algorithms to aggregated glossary data in order to identify potentially dissonant terms. Topic analysis is rarely, if ever, utilized to identify semantically dissonant words in order to streamline inter- and intra-disciplinary communication. Termediator is also unique in its focus on benefiting Information Technology and other adjacent technical fields, as most topic analysis tools tend to focus on either the linguistics field or the English language at large.

Future research should aim to: expand the synonymy tool to include warning list generation, add manual browsing to the polysemy tool, and further increase the dataset of fields where more data was needed to perform adequate polysemous analysis.

7. REFERENCES