SLIDE 1

The Text Categorization – or TC - project is a collaboration between the Journal Descriptor Indexing project in CSB and the Lexical Systems Group in CgSB. Given that JDI is word-based – in particular, characterizing words according to biomedical discipline and high-level category, it seemed a good fit for developing it as a Text Categorization tool along with the other Lexical Systems Group tools. The aim of this talk is to provide a basic understanding of the Text Categorization project methodology, mention research on its use in word sense disambiguation and other applications as time permits, and point you to the Java-based TC tools so you can try it out yourself.

SLIDE 2

The Text Categorization project is concerned with developing tools that categorize text, and also doing research on TC using these tools. In reality, there are currently two types of categorization in this project, known as:
Journal Descriptor Indexing, or JDI, which categorizes text according to Journal Descriptors (JD)
Semantic Type Indexing, or STI, which categorizes text according to Semantic Type (ST).
I'll first be describing JDI and get to STI later.
JDI is concerned with categorizing text according to journal descriptor.

SLIDE 3

What are journal descriptors, or JDs? They are a set of 122 descriptors from the MeSH Vocabulary representing high-level categories, mostly biomedical disciplines, used for indexing MEDLINE journals *per se*. JDs are assigned by a human indexer to the 4100 journals in the training set we use – more about the training set later. The journals and their assigned JDs are part of the List of Serials for Online Users, found in the lsi2007.xml file, which can be ftp'd from the nlmpubs Web site.

SLIDE 4

Here we have examples of information from this serials file. The JID, or Journal Unique Identifier; TA, journal Title Abbreviation; and JDs assigned to three journals – the journal *Transplantation*, with the JD Transplantation; the journal *Pediatric Transplantation*, assigned two JDs Pediatrics and Transplantation, and the *Journal of Pediatric Surgery*, assigned two JDs Pediatrics and Surgery.

SLIDE 5

All 122 JDs are listed, with see and see also references and "includes" notes, and under JD headers, in published form and as a pdf, but these will go away starting 2009, and be replaced by an online counterpart, which can be accessed from PubMed by searching the Journals Database and selecting the "subject terms" link.

SLIDE 6

Here is the online subject list of journals. Beginning with this interface, PubMed users can select journals belonging to a particular JD, and copy them to the PubMed search box in an OR relationship. This will enable them to search, for example, selected Cardiology journals as a search parameter, but this is not the same as searching Cardiology as a topic, because a cardiology document published in *The New England Journal of Medicine* will not be retrieved, since this is not a Cardiology journal. If JDI were applied for searching PubMed, Cardiology as a topic could be a search parameter, not limited to documents in Cardiology journals.

SLIDE 7

So let's get back to how Journal Descriptor Indexing works. To start with, JDI can index a single word, for example the word "transplantation" as shown on this slide. (Just to let you know, I'm going to start with indexing words, then phrases, then MEDLINE documents, and then go on to applications.) The five top-ranked JDs are shown with their scores, as well as the last-ranked of the 122 JDs. The highest-ranked JD is Transplantation, followed by Hematology, Nephrology, Pulmonary Disease (Specialty), and Gastroenterology. The lowest ranked JD, ranked #122, is Speech-Language Pathology. What this means, in simple terms, is that the word "transplantation" is found primarily in Transplantation journals – that is journals assigned the JD Transplantation—in our training set (which I'll get to in a moment), secondarily, the word "transplantation" is found in Hematology journals – that is, journals assigned the JD Hematology, and so forth. The zero score for Speech-Language Pathology means that the word "transplantation" is not found in any of the Speech-Language Pathology journals. I'll get to how these scores were calculated in a moment, and how they lead to indexing of longer text.

SLIDE 8

But first, what is this training set that I've been alluding to? The training set consists of about 3.4 million MEDLINE documents indexed between 1999-2002. JDI requires statistical associations between words in a training set record Title and Abstract and the JDs corresponding to the journal in that training set record. But JDs are not in the MEDLINE record. They are in the NLM serial record from the lsi2007.xml file, I mentioned earlier.

SLIDE 9

As shown here, the JID – Journal Unique Identifier – is in both the training set MEDLINE record, shown first, titled "Combined liver and kidney transplantation in children" from the journal *Transplantation*, and also the serial record beneath it for the

journal *Transplantation*. This JID serves as the link between the journal cited in a MEDLINE record and the journal in the serial record.

SLIDE 10

In fact, one can think of this link as causing the importation of the JD into the MEDLINE training set record. This slide shows the same training set record titled, “Combined liver and kidney transplantation in children,” with the addition of a JD field containing the value Transplantation. Since the MEDLINE record now has access to the JD of the journal, shown here as imported into the MEDLINE record, we can use co-occurrence data, specifically the co-occurrence of words in the TI/AB – namely, the words combined, liver, and, kidney, transplantation, children - with the JD Transplantation – in the indexing of text containing these words, as I’ll show you in a moment.

SLIDE 11

From now on, I’m going to refer to MEDLINE documents, rather than MEDLINE records, but they are the same thing. So let’s go back to our indexing of the word “transplantation,” and explain how the score for the top-ranked JD Transplantation is calculated. The score for the JD Transplantation is the number of documents in the training set - of three years of MEDLINE - in which the TI/AB word “transplantation” co-occurs with the JD Transplantation, divided by the number of training set documents in which the word transplantation occurs in the titles/abstracts. The answer must be a number between 0 and 1 – in the case 0.275691. The score for the JD Hematology is the number of documents in the training set in which the word “transplantation” co-occurs with the JD Hematology, divided by the number of training set documents in which the word transplantation occurs. So you just substitute the JD in this formula to get its score for the word “transplantation.”

SLIDE 12

Here we have the Journal Descriptor Indexing of a different word – the word “kidney” – which was also in our training set document - where Nephrology is the highest ranked JD. The Nephrology score 0.140088 is the number of documents in the training set in which the TI/AB word “kidney” co-occurs with the JD Nephrology, divided by the number of training set documents in which the word “kidney” occurs. Each of the approximately 304,000 words in the training set is indexed in this way. This means the system contains all these words with their associated JDs and scores, ready to be used in some way.

SLIDE 13

Now let’s consider the indexing of a phrase based on the JD indexing of words in the training set. The training set doesn’t contain phrases, but you can index phrases – such as “kidney transplantation” shown here, where the top-five ranked JDs for this phrase are Transplantation, Nephrology, Hematology, and so forth. A JD score for this phrase is the average of the JD score for the word “kidney” and the JD score for the word “transplantation.” Specifically, the score for the top-ranked JD Transplantation, which is 0.178269, is the average of the score for the JD Transplantation, when we indexed the word kidney, and the score for the JD Transplantation, when we indexed the word transplantation. Similarly, the score for the second-ranked JD Nephrology, which is 0.092195, is the average of the score for Nephrology for the word kidney and the score for Nephrology for the word transplantation. In summary, a JD score for a phrase is the average of that JD’s score across the words in the phrase.

SLIDE 14

And that’s basically how JDI works for JD indexing of a text. The average for a particular JD across the words in the text becomes the score for that JD for the entire text. For example, Nephrology will receive a high score for any text with many “kidney” words in it, such as the phrase shown here - “kidney renal nephron glomerulus.” The particularly strong showing for Nephrology compared to the other JDs is due to the fact

that the Nephrology score for each word, when indexed alone, is very high, and therefore the average of these scores must be high as well.

SLIDE 15

It is now possible to perform JD indexing of a document that is outside the training set, such as JDI of the MEDLINE document shown here, based on its title “Kidney transplantation in infants and small children” together with its abstract. The top five JDs are Transplantation, Nephrology, Pediatrics, Hematology, and Urology. Again, the score for each JD is the average of that JD’s score for words in the text. The fact that the “native JDs” of the MEDLINE document are Pediatrics and Transplantation – the JDs for the journal *Pediatric Transplantation* – is totally irrelevant. Only words in the title and abstract are used for JDI of this document.

SLIDE 16

For example, here is the JDI for a title from *The New England Journal of Medicine*, titled, “Pediatric renal-replacement therapy—coming of age.” returning Nephrology, Pediatrics, and Transplantation as the top three JDs. The native JD for *The New England Journal of Medicine* is Medicine. This example is to emphasize this point – that the native JD of a MEDLINE document being indexed does not at all participate in JD Indexing.

SLIDE 17

Internally, the system has word-JD tables representing the JD indexing of each of the 304,000 words in the training set. The scores for an ordered, such as an alphabetical, list of JDs for a word is also called the word-JD vector for that word. Here is part of the word-JD vector for the word “kidney” with scores for four of the 122 JDs – Nephrology, Psychiatry, Psychopharmacology, and Transplantation – in alphabetic order. Note the

scores for Nephrology and Transplantation are relatively high, compared to the scores for Psychiatry and Psychopharmacology.

SLIDE 18

Here is part of the word-JD vector for the word “renal” showing scores for the same JDs. Again, the scores for Nephrology and Transplantation are relatively high, compared to those for Psychiatry and Psychopharmacology.

SLIDE 19

Now we show the word-JD vector for the word “schizophrenia.” Unlike kidney and renal, the scores for Psychiatry and Psychopharmacology are relatively high, compared to those for Nephrology and Transplantation. The zero score for Nephrology is because the word schizophrenia does not appear in any Nephrology journal in the training set.

SLIDE 20

The importance of these word-JD vector examples is to now illustrate a standard measure that compares JD vectors to one another resulting in similarity scores between 0 and 1. The similarity of the JD vector for the word kidney compared to itself is 1.0. The similarity of the JD vector for the word kidney and the JD vector for the word renal is 0.96. But the similarity of the JD vector for the word kidney and schizophrenia is 0.03. The measure we use in our project is the vector cosine coefficient from the well-known textbook by Salton and McGill. We’ll see later why comparing JD vectors is useful.

SLIDE 21

The next three slides show the vector cosine coefficient formula, first for calculating the similarity between the JD vectors of any two words, WORD-i and WORD-j

SLIDE 22

the similarity between the JD vector of any word and the JD vector of any document, WORD-i and DOC-j.

SLIDE 23

and finally the similarity between the JD vectors of any two documents, DOC-i and DOC-j.

SLIDE 24

Now that we know how JDI works – that it's based on word-JD vectors for words, and when you have a text, the scores for a JD for that text are the average scores for that JD across the words in the text – let's talk about Semantic Type Indexing. Semantic types are the set of 135 Semantic Types in the Semantic Network in NLM's UMLS (Unified Medical Language System). Concepts in the UMLS Metathesaurus are assigned one or more STs which semantically characterize those concepts. For example, the concept "aspirin" is assigned the STs Pharmacologic Substance (phsu) and Organic Chemical (orch).

SLIDE 25

Just as the system contains word-JD vectors representing JD indexing for each training set word, the system also contains word-ST vectors representing the semantic type indexing of each training set word. Thus, a text can be indexed according to ST, just as it can be indexed according to JD. An ST score for a text is the average of that ST's score for words in the text. The scores for all the STs comprise the ST vector for the text.

SLIDE 26

How are the word-ST vectors created that are a basis for STI? I like to talk about this because it shows how associations between any two sets that can be categorized in terms of JDs can then be associated with each other, paving the way for new categorizations in terms of one of the sets. Regarding JD vectors, when there is an X-JD vector and Y-JD vectors, an X-Y vector can be created. Here is an example of how a word-JD vector and semantic type-JD, or ST-JD vectors, can result in a word-ST vector. Based on our training set, the word "transporting" has a JD vector (JD1, JD2, etc., with scores). Let's say that each semantic type also has a JD vector, such as the JD vector for the semantic type Cell Function and the semantic type Health Care Activity, as depicted on this slide. These are the semantic types associated with two of the Metathesaurus senses of the word transporting in our WSD study, namely biological transport (a cell function) and patient transport (a health care activity), which I'll discuss later.

SLIDE 27

Using the vector cosine coefficient, the similarity between the JD vector for the word transporting and the JD vector for the semantic type Cell Function is computed as 0.7252; the similarity between the JD vector for the word transporting and the JD vector for the semantic type Health Care Activity is 0.3890. As you can see below, we have the start of the word-ST vector for the word transporting. We just have to do the remainder of the semantic types in the same way, and we will have the complete word-ST vector for this word. We then create word-ST vectors for all words, which form the basis for performing ST indexing. By the way, the higher score for Cell Function compared to Health Care Activity indicates that the predominant sense for transporting in the training set is that of biological transport rather than patient transport. As we will see later, these sorts of comparative score form the basis for our word sense disambiguation application. But our fundamental problem is coming up with JD vectors for the semantic types. How can we represent a semantic type in order to do the JD indexing of it?

SLIDE 28

Our answer to representing semantic types is to create “semantic type documents,” or ST documents. The contents of an ST document are one-word Metathesaurus strings belonging to the semantic type. Shown in this slide are the ST documents for the semantic types Cell Function and Health Care Activity. The words in the TI field belong only to the semantic type. Those in the AB field belong to the semantic type and other semantic types as well. The distinction between TI and AB words isn’t used, but it can potentially can be. (The actual implementation in our TC system isn’t done in quite this way, but it is useful to portray an ST document as if it were a MEDLINE document for explanation purposes.)

SLIDE 29

This slide contains the same table as before, with the word-JD vector for transporting and two of the ST-JD vectors, except the surrogates for the STs Cell Function and Health Care Activity are shown as the Cell Function (celf for short) and the Health Care Activity (hlca for short) documents. Again, when we have the word-ST vectors for all the words in the training set, we can do semantic type indexing of text based on word-ST vectors of words in the text in the same manner as we do JD indexing based on word-JD vectors of words in the text.

SLIDE 30

Research has been published on ST indexing as a tool for disambiguating text. Disambiguation is a major challenge in natural language processing, such as that performed by MetaMap, on which the automated Medical Text Indexer is based. STI was used for disambiguating 45 ambiguous strings from NLM’s WSD collection, which had been disambiguated by humans as the gold standard. The number of instances for each ambiguity ranged from 3 to 67, with an average of 54. Instances for which “None of the Above” was the gold standard were ignored, since neither STI nor the baseline method to which it was compared was designed to return this answer. The study was published in January 1, 2006, issue of *JASIST* (Humphrey SM, Rogers WJ, Kilicoglu H,

Demner-Fushman D, and Rindflesch TC. Word sense disambiguation by selecting the best semantic type based on Journal Descriptor Indexing: preliminary experiment. *J Am Soc for Inf Sci and Technol*. 2006 Jan 1;57(1):96-113. Erratum in: *J Am Soc Inf Sci Technol*. 2006 Mar;57(5):726.)

SLIDE 31

For example, the ambiguity "transport" has two meanings: "Biological Transport" assigned the ST Cell Function (celf) and "patient transport" assigned the ST Health Care Activity (hlca). The STI methodology can analyze text, such as a MEDLINE document, containing an ambiguous string and determine which of the STs assigned to that string by UMLS receives a higher score for that text, which then returns the associated meaning, presumed to apply to the ambiguity itself. If celf ranks higher than hlca, the meaning is Biological Transport; if hlca ranks higher than celf, the meaning is Patient Transport.

SLIDE 32

This sample input corresponding to title and abstract of PMID 9674486 contains the ambiguity "transporting" (a variant of transport) in the last sentence, "This practice averts the potential complications associated with transporting critically ill patients." When a system like MetaMap encounters such an ambiguity, it needs to know the correct meaning. We as humans can easily disambiguate the word "transporting," choosing the correct ST of hlca (for Health Care Activity) over the ST celf (for Cell Function). Automatic STI also successfully performed this disambiguation, according to the higher score for the ST hlca for this document, compared to celf.

SLIDE 33

One of the issues is the context of the ambiguity, which may be the one sentence with the ambiguity, all sentences with the ambiguity, the entire MEDLINE document, or involve a rule to use the entire document when the ambiguous sentence has fewer words than some

threshold. In this study, STI achieved an overall average precision of 0.7873 compared to 0.2492 for a baseline method known as MeSH Frequency. The baseline method involves automatically matching each candidate concept for an ambiguity to a MeSH synonym if there is one. The concept matching the MeSH synonym with the highest frequency count in MEDLINE is returned as the answer. If some concept has no MeSH synonym, then it has no chance of being the answer. For example, if there are two candidates for an ambiguity, and the correct one has no corresponding MeSH synonym, then the other concept wins for all instances of the ambiguity in the collection, even if the first candidate is the correct answer for most or even all the instances.
STI continues to be investigated for WSD in NLP applications related to the Indexing Initiative and Semantic Knowledge Representation.

SLIDE 34

A "JDI method" is one of the methods investigated in the Subheading Attachment Project, which developed an automatic subheading attachment module for NLM's Medical Text Indexer (MTI). Before this, MTI recommended main headings but not subheadings attached to them. The JDI method produces a ranked list of the top five subheadings for a text to be indexed, and is combined with other methods in the project. The method depends on the fact that the training set used for JDI contains not only word-JD vectors, but also subheading-JD vectors. That is, training set documents contain not only titles and abstracts, but also the MeSH indexing, and a subheading-JD vector is calculated similarly to a word-JD vector. The score for a JD is the number of documents in the training set in which the subheading co-occurs with a JD, divided by the number of documents in which the subheading occurs. For this application, we aren't really interested in the JD vector for the subheading as an end-product, but only for comparing this vector against a word-JD vector. Let's say you have the JD vector for the word surgical and the JD vectors for the subheadings surgery, blood supply, and abnormalities. The similarity between the JD vector for surgical and the JD vector for the subheading surgery is 0.9613; the similarity between the JD vector for surgical and the JD vector for the subheading blood supply is 0.8075; and the similarity between the JD vector for

surgical and the JD vector for the subheading abnormalities is 0.7804. So the subheading most correlated with the word surgical is surgery. If you do this comparison between the JD vector for surgical and JD vector for all the other 80 subheadings, the result will be a word-subheading vector for surgical, where the scores for the subheadings (SHs) are the similarity scores. Shown here are the three subheadings in the surgical-SH vector. Again, the principle is similar to creating word-ST vectors described earlier. In this case, when you have a word-JD vector and subheading-JD vectors, you can create a word-subheading vector, where the score for each subheading is the similarity between the word-JD vector and the JD vector for that subheading. Doing this for every word in the training set produces 304,000 word-subheading vectors, which can be used for MeSH subheading indexing of text, just as word-ST vectors can be used for ST indexing of text, and just as word-JD indexing can be used for JD indexing of text.

SLIDE 35

This table shows the result of MeSH subheading indexing (using word-SH vectors) produced by the JDI method for the title, “The role of surgical decompression for diabetic neuropathy.” The top five SHs of the title-subheading vector for this title are shown in the SHs column - blood supply, complications, etiology, physiopathology, and surgery. Columns under the words show the scores for the word-SH vectors for these subheadings. For example, the score for the subheading surgery in the surgical-SH vector is 0.9613; the score for the subheading surgery in the decompression-SH vector is 0.7455; the score for the subheading surgery in the diabetic-SH vector is 0.1963; and so on. The scores for the title were produced by averaging the scores for each subheading across the words, as shown in the avg column, and the rank for each subheading is shown in the rank column. As shown, the subheading surgery has the highest average across the words, and is therefore ranked number 1, etiology second, complications third, physiopathology fourth, and blood supply fifth. This statistical correlation between a text and MeSH subheadings is combined with other methods to suggest subheading assignments to MTI main heading recommendations. Using this example, if MTI recommended the main heading Diabetic Neuropathies based on this title, this method would be used for suggesting the assignment

of the subheading surgery to this main heading. This research is included in a submission to *Journal of Biomedical Informatics*, with first author Aurélie Névéol.

SLIDE 36

Other research we are exploring involves the ability of JDI to categorize a document as being in the genetics domain or not, as a step in gene symbol disambiguation. A symbol in a document determined to be in the genetics domain would more likely be a gene symbol than if it occurs in a non-genetics document. Researchers Andrej Kastrin and Dimitar Hristovski of Ljubljana University in Slovenia developed a document classifier based on whether there is a statistically significant difference between observed frequencies of MeSH descriptors in the full MEDLINE corpus and a subset genetic domain corpus. A decision score indicating genetics or not genetics could then be computed for a MEDLINE document according to its MeSH indexing terms. Their forthcoming AMIA 2008 paper reports that their classifier achieved predictive accuracy of 0.91 with 0.93 precision and 0.64 recall. In their study, experts annotated two sets of 100 MEDLINE documents as to whether they were in the genetics domain or not, which they were kind enough to forward us. We used one set as our training set (the same set that Kastrin and Hristovski used as their training set to tune their classifier), and ran JDI and STI limiting to certain genetics JDs and STs. We selected as our four genetics JDs: Genetics; Genetics, Behavioral; Genetics, Medical; and Molecular Biology. As genetics STs, we selected two possible groups. The first is Gene or Genome; Genetic Function; and Nucleotide Sequence. The second adds two more STs: Nucleic Acid, Nucleoside, or Nucleotide; and Molecular Biology Research Technique. We recorded the highest rank of the JDs or the STs or the combined JDs and STs, as well as the results of two methods we use, one based on word count and the other based on document count, and evaluated if we could use these ranks as a cut-off for identifying genetics vs. non-genetics documents. There were 15 permutations, for example, one permutation used the highest rank of 4 JDs + 3 STs with either the wc or the dc method.

SLIDE 37

As it turned out, the best set of JDs/STs was the STs consisting of three STs Genetic Function, Gene or Genome, and Nucleic Acid Sequence, in either the word count or the document count method. For example, this slide shows two documents and ranks of three genetics STs for the word count method and the document count method. The first document had been annotated as being in the genetics domain. The system ranked Gene or Genome first in both methods, which is a good result, as it agrees with the human annotator, that is, a true positive. The second document had been annotated as not being in the genetics domain. Gene or Genome is ranked 75 for the word count method and 77 for the document count method, which, again, is a good result, that is, a true negative. The question in this research is, where should the cut-off be, somewhere between 1 and 75? That is, what is the best threshold rank, above which documents would be considered by the system to be in the genetics domain, in order to agree with human annotators?

SLIDE 38

As seen in this table, generated by Mehmet Kayaalp, the best cut-off was at rank 13, giving optimum performance. In other words, if any one of the three genetics STs Genetic Function, Gene or Genome, or Nucleic Acid Sequence, ranked 13 or better, the document was categorized by the system as being in the genetics domain. This cut-off achieved optimum performance on the training set - predictive accuracy of 0.94, recall 0.77, precision 0.94, and F-score 0.85. This research is still in progress. We shifted gears, and decided to incorporate into the research an automated method to select the JDs and STs in the first place out of all the JDs and STs, rather than rely on a human selecting them. We used our methodology on the training set to determine the JD/ST criteria and threshold, and applied them to the test set. Our preliminary results show encouraging performance of our classifier (which is based on text) in comparison to the performance of the classifier of Kastrin and Hristovski (which is based on MeSH indexing).

We note that, as far as domain of interest application is concerned, JDI has been used for several years by SemRep as a pre-processing step to increase accuracy by identifying MEDLINE documents in the molecular genetics domain before NLP begins.

SLIDE 39

We also are contemplating research involving JD vector similarity.
An example of research based on similarity between word-JD vectors in the training set has the goal of automatic creation of stopword lists; our current stopword list was developed empirically. This is done by comparing the JD vector for the quintessential stopword THE to all the other words in the training set. In theory, a word with a JD vector similar to THE – with all low, gradually decreasing scores when viewed by score) would likely be a good stopword as well. Here we show a result of similarity of the word-JD vector for the word THE to its most similar words. Similarity of THE to itself is, of course 1.0. To AND is 0.9998. To FOR is 0.9977. To WITH is 0.9970. The most dissimilar word to THE in the training set is COMLEX, which is an acronym for Comprehensive Osteopathic Medicine Licensing Examination, exclusively associated with the JD Osteopathy and with a score of 0.0028.

SLIDE 40

Another avenue of research involves comparing JD vectors of different indexing terms to the same MEDLINE document. In theory, the more similar a term JD vector is, to a MEDLINE document JD vector, the more descriptive that term is of the MEDLINE document, and by contrast, a JD vector for a term that is very dissimilar to a MEDLINE document JD vector, would not be a good descriptor for the document. Thus, an indexing term assigned to a document – whether as a recommendation from an automated indexing system such as MTI or humanly-assigned – might be detected as an outlier because of the great dissimilarity of the term's JD vector to the JD vector of the document being indexed. The impetus for the outlier detector was the coming to our attention that MTI was recommending the indexing term Stupor resulting from the word "unresponsive" in a

MEDLINE document even when the document was referring to unresponsive cells. This recommendation would be considered a blooper for indexing such a document. As shown in this slide, the similarity is only about 0.2 between the JD vector for the term "Stupor" and for the title/abstract of MEDLINE document referring to human intestinal epithelial cells that are unresponsive to Toll-like receptor2-dependent bacterial ligands. However, the similarity between the other indexing terms is much higher, for example, 0.9 for the recommendation Toll-Like Receptor 2. So, Stupor stands out as an inappropriate indexing term compared to the others, based on vector similarity. We are investigating if this phenomenon can be used for detecting other such blooper recommendations. An example of blooper detection in human indexing involved the assignment of the MeSH term Deception (a social behavior term) to documents describing the bacterium Myxococcus xanthus as a cheater. A successful blooper detector would compute a low similarity of 0.14 between the JD vector for the term "Deception" and the JD vector for this document, compared to a high similarity of 0.82 between the JD vector for the term "Myxococcus xanthus" and the document.

SLIDE 41

Here are some other ideas on further studies involving JDI.

Evaluate JDI by running JDI on MEDLINE documents from a journal, thus creating a journal-JD vector by averaging the JD scores across documents in the journal, and use as criterion of success whether the native JD of the journal is ranked high in the journal-JD vector. Evaluate STI using MeSH indexing to determine the gold standard meaning.

Creating specialty subsets of general medical journals, such as *The New England Journal of Medicine* or *JAMA*, or the journal *Science*. Or partitioning any large, varied collection into specialties for users who would like to be alerted to relevant material in their specialty or some intersection of specialties. Or partitioning all of MEDLINE so that specialities can be a PubMed search parameter.

JDI is word-based. Some have suggested that it be phrase-based, or that we consider variants of a word to be a single word (combining variants was actually an unimplemented option in the original Lisp system).

One could possibly expand JDI beyond biomedicine by using LC call numbers as JDs, and developing a training set from collections representing all subjects. Since many biomedical journals also have LC call numbers, can try using them instead of JDs in the current system. This would require buying some files from the Library of Congress (serials file with corresponding LC classification numbers, and file containing LC subclasses with corresponding descriptions). Acquiring a training set in many subjects could be done by collaborating with search vendors who maintain many bibliographic files, such as Dialog (owned by Reuters) or Wilson Web (owned by H.W. Wilson) which maintain hundreds of databases from a broad scope of disciplines.

SLIDE 42

Shown here is a successful example of creating a journal-JD vector to corroborate the humanly-assigned JDs for a new journal titled *Bioinspiration & Biomimetics*. JDI corroborated humanly assigned JDs Biology and Biomedical Engineering. JDI was performed on 20 MEDLINE documents from this journal, and the scores for the same JD were averaged across these documents. Mehmet Kayaalp used the TC command line tools that generated the JDI of the documents in a journal, and wrote the scripts that retrieved the documents from PubMed and that did the averaging of the JD scores of the documents from a journal, resulting in the journal-JD vector. We did this for 126 new journals for 2007. Hopefully, we can continue this research on a larger sample from many more journals, and then evaluate the results against the humanly-assigned JDs. This example shows a potential problem area, which is that sometimes the gold standard isn't quite right. As we understand it, this journal deals with biology to solve engineering problems, whereas Biomedical Engineering deals with engineering to solve biological problems.

SLIDE 43

Most of the JDI and STI in this talk can be done by using the TC Web Tools at the TC Web site http://specialist.nlm.nih.gov/tc. For extensive research, command line tools can

be used. TC tools and applications are freely distributed with open source code, 100% in JAVA, running on different platforms. It is one complete package with documentation and support, and provides Java APIs, command line tools and Web tools. You can click on Documents at the TC Web site for links to our publications, including the WSD paper. A new release, TC 2008, has just been completed which adds functionality, and creates a new training set from MEDLINE documents, and ST documents from the Metathesaurus, and the word-JD and word-ST vectors derived from them. The TC 2007 release uses a fixed training set and vectors that were given to it from the Lisp system I created. We also hope to develop features to facilitate research, for example the ability to test different ST documents and stopword lists developed outside the system. The Java system was developed by Chris Lu and authorized by Allen Browne, both of the Lexical Systems Group; Willie Rogers, working under the Indexing Initiative project in CgSB, is a collaborator on the TC project.

SLIDE 44

This slide shows use of command line. The MEDLINE tokenizer command, mlt2007, shown here, takes as input a file in MEDLINE format, and outputs a file of tokenized text of the TI and AB. The JDI command, jdi2007, shown here, takes as input the tokenized file, and outputs a file of the JDI result of all 122 JDs. The STI command, sti2007, shown here, takes as input the tokenized file, and outputs a file of the STI result of only three specified STs Genetic Function, Gene or Genome, and Nucleic Acid Sequence. Command lines are described at the TC Web site by selecting Documents and then the command of interest.

SLIDE 45

Here are some statistics comparing the 2007 and 2008 TC releases. 2007 has about 4,100 journals, 1.4 million MEDLINE documents indexed between 1999-2001, and 304,000 unique words in their titles and abstracts. 2008 has about 5,200 journals, nearly 2 million MEDLINE documents indexed between 2005-2007, and about 397,000 unique words in

their titles and abstracts. Chris Lu is primary author of an interesting forthcoming paper for 2008 AMIA on a fast, effective, and accurate method to compute a similarity index - between 0 and 1 - representing the similarity between two sets of vectors from training sets from different years, including different time periods and different durations of years. A high similarity index serves to validate the new training set.

SLIDE 46

This is a list of some of the challenges in our research. Normalization of counts is essential; otherwise, high-frequency words and journals with many document skew the results. Practically all applications require the computation of thresholds. The success of STI depends on the "best" ST documents; perhaps the best ST documents should be comprised of words that purely belong to the semantic type. In some cases, perhaps only one word is needed; for example, might an ST document consisting of only the word "food" be sufficient for the ST Food? There is ambiguity in Metathesaurus ST assignments; perhaps words in an ST document should be restricted to being in a single semantic group. Is there a way to develop a stopword list automatically, and does that mean that stopwords can be dataset-specific? There are some JD issues, although by and large, specialties don't change that much over time. For example, Gynecology and Obstetrics are separate JDs, and the journal in the training set *American Journal of Obstetrics and Gynecology* is assigned both JDs. This means that words in a document in the training set from this journal are associated with both JDs, whether the document is in the field of Obstetrics or Gynecology. For example, the word "pregnancy" (an obstetrics word) is associated with Gynecology (as well as with Obstetrics), and the word "gynecology" (a gynecology word) is associated with Obstetrics (as well as with Gynecology). This causes these JDs to have similar scores in the results of JDI, whether the text is in the field of Obstetrics or Gynecology. Another problematic journal is *Annals of Thoracic and Cardiovascular Surgery*, which has JDs Cardiology, Pulmonary Disease (Specialty), Surgery, and Vascular Diseases, and therefore the words in a training set title from this journal, such as, "Middle mediastinal thymoma of unusual pathologic type," would be associated with each of these JDs. A general solution might be to

remove from the training set those journals assigned multiple JDs where each JD does not generally apply to all documents in the journal. And finally, an important challenge is to establish a testing suite in order to evaluate research in the Text Categorization project.

SLIDE 47

Just to illustrate creating specialty subsets. There's a real-world example on the Web site of American Medical Association publications. Published studies in *JAMA* and AMA *Archives* journal have been categorized by Topic Collections since January 1998. For example, a geriatrician can click on Aging/Geriatrics as a link to documents in *JAMA* and *Archives* journals, such as *Archives of Internal Medicine*, beginning with the most recent issue.

SLIDE 48

For the Web site of the American Academy of Pediatrics, editors have been categorizing published studies in the journal *Pediatrics*, since January 1997, according to subspecialties similar to JDs. For example, a pediatric oncologist can select Tumors as a link to full-text documents in this journal on the subject of childhood cancer, beginning with the most recent issue.

SLIDE 49

Another real-world example is the Web site of the journal *Science*, published by the American Association for the Advancement of Science. Since 1996 editors have been categorizing published studies in the journal *Science*, according to Science Subject Collections. For example, a meteorologist can select Atmospheric Science as a link to documents in this journal on this subject, ordered by most recently published. There is also a search box for entering keywords in a selected Collection.

SLIDE 50

Suppose a topic of interest is inflammation in the field of cardiology. A simple PubMed search strategy would be to intersect cardiology journals with the search term inflammation. But what if there are studies on this topic published in a non-cardiology journal, like the *New England Journal of Medicine*, shown here. You wouldn't get it. What's needed is an indexing term for the cardiology parameter, so you don't have to limit the search to cardiology journals, nor do you need to express cardiology with keywords, which can be quite labor intensive. If such specialty indexing existed, you could even intersect specialties, like cardiology intersected with allergy and immunology, and use this retrieval as a current awareness search sent to your mailbox periodically. JDI can index text by such broad categories, and has the potential to allow searching according to them.

SLIDE 51

This and the next two slides were not part of the presentation on June 27, 2008, but they may be of interest, as they show how JDI can be extended to perform MeSH indexing by assigning mainheadings. The training set used for JDI contains not only word-JD vectors and subheading-JD vectors, as we've seen before, but also mainheading-JD vectors. The score for a JD in an MH-JD vector is the number of documents in the training set in which the MH co-occurs with a JD, divided by the number of documents in which the MH occurs. Using the same methodology described earlier, we can create, from a word-JD vector and the MH-JD vectors, a word-MH vector. The two top-scoring MHs for the MH vector for the word transporting are Carrier Proteins and Isoenzymes. Also shown are the scores and ranks for the MHs Protein Transport and Transportation of Patients, and the 15 MHs with 0 score. If we have word-MH vectors for all words in the training set, we can perform automated indexing by mainheadings.

SLIDE 52

Here are some similar data for MH vectors for the words surgical and decompression. The best MHs for the word surgical are Fibrin Tissue Adhesive; Homeostasis, Surgical; and Postoperative Complications. Decompression, Surgical has a score that ranks it about 191th. The best MHs for the word decompression are Decompression, Surgical; Odontoid Process; and Nerve Compression Syndromes. When we index the text "surgical decompression," by averaging the MH scores in the vectors for the two words, the top-ranked MH is Decompression, Surgical, so the score for Decompression, Surgical in the surgical-MH vector was high enough to maintain the top rank for this MH for the "surgical decompression" text.

SLIDE 53

We did some experiments on JDI based MH indexing for 16 words. This slide shows the top-scoring MH for each word. Considering there were about 20,000 MHs in the word-MH vectors, these results don't seem too bad. Remember, this technique doesn't use natural language processing. It's all discipline-based – comparing word-JD vectors with MH-JD vectors. However, the length of the vectors may pose a technical problem if fast processing is desired in an application; resolving this problem would require investigation.

SLIDE 54

The people I've worked with over the years are listed in this slide. Those on the research side, in the Lister Hill Center, are listed in the left-hand column. In the right-hand column are the people in Library Operations responsible for assigning and making available Journal Descriptors.