Automatic Classification of PubMed Abstracts with Latent Semantic Indexing

J Robert Adams, Steven Bedrick

Center for Spoken Language Understanding
Oregon Health and Science University

BioASQ Challenge 2A, 2014

Outline

Outline

Proposed Approach

The goal of BioASQ Task 2a is to automatically assign MeSH index headings to un-tagged MEDLINE abstracts.

- Approach the problem from a document clustering perspective.
- Similar documents often share MeSH terms.
- Use Latent Semantic Analysis (LSA) to identify semantically “similar” articles to an unlabeled (‘query’) abstract.
- Use the human-assigned MeSH descriptors of the similar abstracts to build a set of candidate descriptors.

Proposed Approach

The goal of BioASQ Task 2a is to automatically assign MeSH index headings to un-tagged MEDLINE abstracts.

- Approach the problem from a document clustering perspective.
- Similar documents often share MeSH terms.
- Use Latent Semantic Analysis (LSA) to identify semantically “similar” articles to an unlabeled (‘query’) abstract.
- Use the human-assigned MeSH descriptors of the similar abstracts to build a set of candidate descriptors.

Proposed Approach

The goal of BioASQ Task 2a is to automatically assign MeSH index headings to un-tagged MEDLINE abstracts.

- Approach the problem from a document clustering perspective.
- Similar documents often share MeSH terms.
- Use Latent Semantic Analysis (LSA) to identify semantically “similar” articles to an unlabeled (‘query’) abstract.
- Use the human-assigned MeSH descriptors of the similar abstracts to build a set of candidate descriptors.

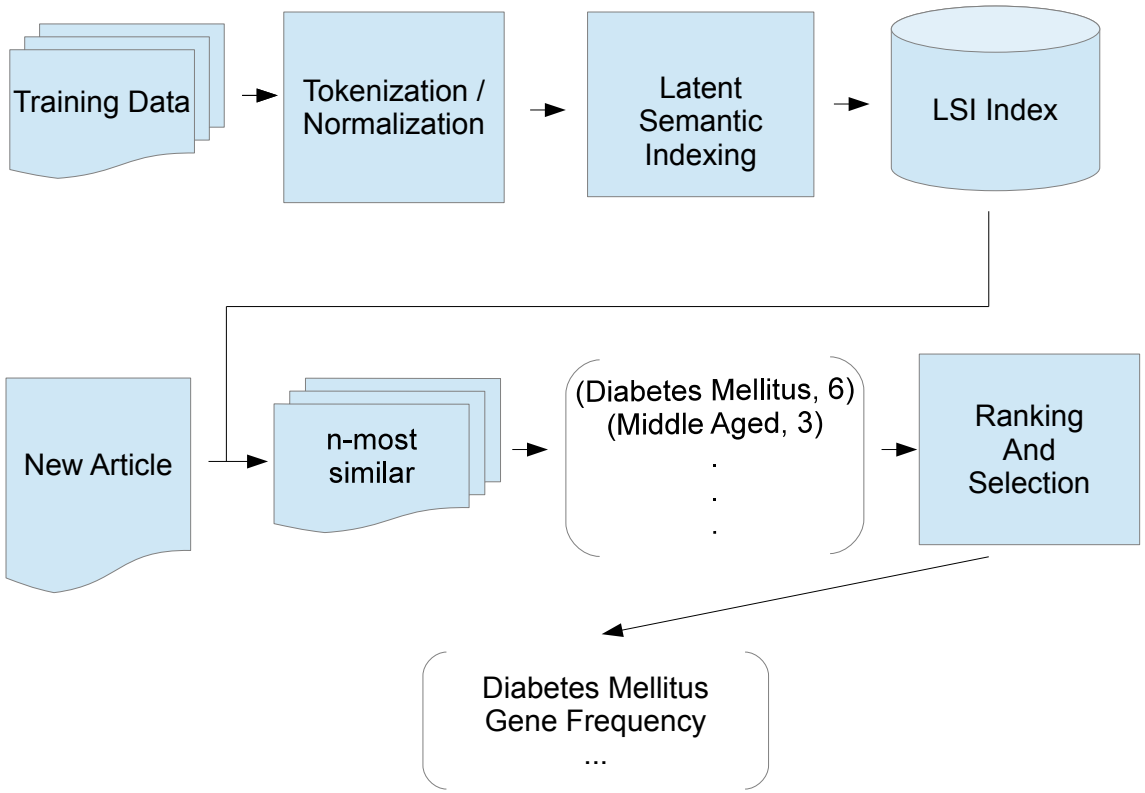
Proposed Approach

The goal of BioASQ Task 2a is to automatically assign MeSH index headings to un-tagged MEDLINE abstracts.

- Approach the problem from a document clustering perspective.
- Similar documents often share MeSH terms.
- Use Latent Semantic Analysis (LSA) to identify semantically “similar” articles to an unlabeled (‘query’) abstract.
- Use the human-assigned MeSH descriptors of the similar abstracts to build a set of candidate descriptors.

Outline

System Overview



Outline

Choosing the Training Data Set

Due to the large amount of potential training data and the changing nature of the MeSH tree, we chose to narrow our selection of training data.

- Only articles included in the list of 1,993 journals which BioASQ identified as having “small average annotation periods”
- Only descriptors which appear in the 2014 edition of MeSH.
- Trained on a subset of the provided *Training Set v.2014b* restricted to articles from 2005 and later ($\approx 1.5M$ abstracts)
- When experimenting with metavariables, we used a randomly assigned 90/10 learning/validation split.
- When classifying new articles we used a model based on the entire training set.

Choosing the Training Data Set

Due to the large amount of potential training data and the changing nature of the MeSH tree, we chose to narrow our selection of training data.

- Only articles included in the list of 1,993 journals which BioASQ identified as having “small average annotation periods”
- Only descriptors which appear in the 2014 edition of MeSH.
- Trained on a subset of the provided *Training Set v.2014b* restricted to articles from 2005 and later ($\approx 1.5M$ abstracts)
- When experimenting with metavariables, we used a randomly assigned 90/10 learning/validation split.
- When classifying new articles we used a model based on the entire training set.

Choosing the Training Data Set

Due to the large amount of potential training data and the changing nature of the MeSH tree, we chose to narrow our selection of training data.

- Only articles included in the list of 1,993 journals which BioASQ identified as having “small average annotation periods”
- Only descriptors which appear in the 2014 edition of MeSH.
- Trained on a subset of the provided *Training Set v.2014b* restricted to articles from 2005 and later ($\approx 1.5M$ abstracts)
- When experimenting with metavariables, we used a randomly assigned 90/10 learning/validation split.
- When classifying new articles we used a model based on the entire training set.

Choosing the Training Data Set

Due to the large amount of potential training data and the changing nature of the MeSH tree, we chose to narrow our selection of training data.

- Only articles included in the list of 1,993 journals which BioASQ identified as having “small average annotation periods”
- Only descriptors which appear in the 2014 edition of MeSH.
- Trained on a subset of the provided *Training Set v.2014b* restricted to articles from 2005 and later ($\approx 1.5M$ abstracts)
- When experimenting with metavariables, we used a randomly assigned 90/10 learning/validation split.
- When classifying new articles we used a model based on the entire training set.

Choosing the Training Data Set

Due to the large amount of potential training data and the changing nature of the MeSH tree, we chose to narrow our selection of training data.

- Only articles included in the list of 1,993 journals which BioASQ identified as having “small average annotation periods”
- Only descriptors which appear in the 2014 edition of MeSH.
- Trained on a subset of the provided *Training Set v.2014b* restricted to articles from 2005 and later ($\approx 1.5M$ abstracts)
- When experimenting with metavariables, we used a randomly assigned 90/10 learning/validation split.
- When classifying new articles we used a model based on the entire training set.

Outline

LSA: Motivation

- Using LSA, one may perform vector- space retrieval on a low-rank approximation of a term-document matrix, in which “related” words end up grouped together.
- The combination of dimensionality reduction and semantic grouping seemed to make LSA a natural fit for the problem of computing document similarity for automatic indexing.

LSA: Motivation

- Using LSA, one may perform vector- space retrieval on a low-rank approximation of a term-document matrix, in which “related” words end up grouped together.
- The combination of dimensionality reduction and semantic grouping seemed to make LSA a natural fit for the problem of computing document similarity for automatic indexing.

Latent Semantic Analysis

- LSA produces a matrix approximation using singular value decomposition (SVD). SVD effectively "splits" a term-document matrix X into three new matrices, U, S, and V, which may be multiplied together in order to re-create the original matrix ($X = USV^T$).

$$\underset{m \times n}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}} = \underset{m \times r}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}} \underset{r \times r}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}} \underset{r \times n}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}$$

- The decomposition can be used to create lower dimensional approximations of the original matrix.

Latent Semantic Analysis

- LSA produces a matrix approximation using singular value decomposition (SVD). SVD effectively "splits" a term-document matrix X into three new matrices, U, S, and V, which may be multiplied together in order to re-create the original matrix ($X = USV^T$).
- $$\underset{m \times n}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}} = \underset{m \times r}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}} \underset{r \times r}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}} \underset{r \times n}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}$$
- The decomposition can be used to create lower dimensional approximations of the original matrix.

Latent Semantic Analysis

- LSA produces a matrix approximation using singular value decomposition (SVD). SVD effectively "splits" a term-document matrix X into three new matrices, U, S, and V, which may be multiplied together in order to re-create the original matrix ($X = USV^T$).
- $$\underset{m \times n}{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & \\ \vdots & \vdots & \ddots & \\ x_{m1} & & & x_{mn} \end{pmatrix}} = \underset{m \times r}{\begin{pmatrix} u_{11} & \cdots & u_{1r} \\ \vdots & \ddots & \\ u_{m1} & & u_{mr} \end{pmatrix}} \underset{r \times r}{\begin{pmatrix} s_{11} & 0 & \cdots \\ 0 & \ddots & \\ \vdots & & s_{rr} \end{pmatrix}} \underset{r \times n}{\begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \ddots & \\ v_{r1} & & v_{rn} \end{pmatrix}}$$
- The decomposition can be used to create lower dimensional approximations of the original matrix.

Training: Tokenization and Normalization

Simple tokenization was implemented with the Python Natural Language Toolkit (NLTK) library.[1]

- Sentence tokenization via Punkt
- Word tokenization using standard NLTK word tokenizer.
- Removed members of the NLTK English stop word list.

[1] http://www.nltk.org/

Training: Tokenization and Normalization

Simple tokenization was implemented with the Python Natural Language Toolkit (NLTK) library.[1]

- Sentence tokenization via Punkt
- Word tokenization using standard NLTK word tokenizer.
- Removed members of the NLTK English stop word list.

[1] http://www.nltk.org/

Training: Tokenization and Normalization

Simple tokenization was implemented with the Python Natural Language Toolkit (NLTK) library.[1]

- Sentence tokenization via Punkt
- Word tokenization using standard NLTK word tokenizer.
- Removed members of the NLTK English stop word list.

[1] http://www.nltk.org/

Training: Building the LSI Index

We created our LSI index using Gensim[2]

- Create a term-document matrix representation of our training corpus.
- Transform the frequency counts into normalized Term Frequency-Inverse Document Frequency (TF-IDF) scores.
- Create LSI index of our corpus with the first 200 eigenvalues of the decomposed matrix.

[2] http://radimrehurek.com/gensim/

Training: Building the LSI Index

We created our LSI index using Gensim[2]

- Create a term-document matrix representation of our training corpus.
- Transform the frequency counts into normalized Term Frequency-Inverse Document Frequency (TF-IDF) scores.
- Create LSI index of our corpus with the first 200 eigenvalues of the decomposed matrix.

[2] http://radimrehurek.com/gensim/

Training: Building the LSI Index

We created our LSI index using Gensim[2]

- Create a term-document matrix representation of our training corpus.
- Transform the frequency counts into normalized Term Frequency-Inverse Document Frequency (TF-IDF) scores.
- Create LSI index of our corpus with the first 200 eigenvalues of the decomposed matrix.

[2] http://radimrehurek.com/gensim/

Outline

Create Query from Test Document

- Tokenization / stopword removal.
- Project the document into the lower dimensional space.
- Calculate cosine similarity against all of our training documents.
- Candidate set is the MeSH terms of the 20 most similar documents.

Create Query from Test Document

- Tokenization / stopword removal.
- Project the document into the lower dimensional space.
- Calculate cosine similarity against all of our training documents.
- Candidate set is the MeSH terms of the 20 most similar documents.

Create Query from Test Document

- Tokenization / stopword removal.
- Project the document into the lower dimensional space.
- Calculate cosine similarity against all of our training documents.
- Candidate set is the MeSH terms of the 20 most similar documents.

Create Query from Test Document

- Tokenization / stopword removal.
- Project the document into the lower dimensional space.
- Calculate cosine similarity against all of our training documents.
- Candidate set is the MeSH terms of the 20 most similar documents.

Assigning Descriptors to Our New Document

We developed a simple scoring algorithm to rank the candidate descriptors based on the following assumptions:

1. All else being equal, a MeSH term associated with a *more* similar document should have a greater contribution to the score than a heading from a *less* similar document.
2. Terms which appear *more frequently* in neighboring documents are better candidates than those which only occur a single time.
3. This second point is mediated by the fact that some MeSH headings, such as the check tag “Human” are much more frequent in the corpus than others, so neighbors sharing one of these contributes less information than files sharing a more obscure header.

Assigning Descriptors to Our New Document

We developed a simple scoring algorithm to rank the candidate descriptors based on the following assumptions:

1. All else being equal, a MeSH term associated with a *more* similar document should have a greater contribution to the score than a heading from a *less* similar document.
2. Terms which appear *more frequently* in neighboring documents are better candidates than those which only occur a single time.
3. This second point is mediated by the fact that some MeSH headings, such as the check tag “Human” are much more frequent in the corpus than others, so neighbors sharing one of these contributes less information than files sharing a more obscure header.

Assigning Descriptors to Our New Document

We developed a simple scoring algorithm to rank the candidate descriptors based on the following assumptions:

1. All else being equal, a MeSH term associated with a *more* similar document should have a greater contribution to the score than a heading from a *less* similar document.
2. Terms which appear *more frequently* in neighboring documents are better candidates than those which only occur a single time.
3. This second point is mediated by the fact that some MeSH headings, such as the check tag “Human” are much more frequent in the corpus than others, so neighbors sharing one of these contributes less information than files sharing a more obscure header.

Assigning Descriptors Part 2

- For any MeSH header m in our set of candidates, we define a weighted frequency $f(m)$

$$f(m) = \sum_{i=1}^{n} e(i) \cdot s_i \ . \tag{1}$$

Where:

$$e(i) = \begin{cases} 1 & \text{if } m \in M_i \\ 0 & \text{otherwise} \ . \end{cases} \tag{2}$$

- Inverse document frequency $idf(m)$ over the training corpus:

$$idf(m) = \log(\frac{N}{1+C}) \tag{3}$$

- Final score is:

$$score(m) = f(m) \cdot idf(m) \tag{4}$$

- Lower threshold of 1.5, return the highest scored MeSH descriptors (max 12).

Assigning Descriptors Part 2

- For any MeSH header m in our set of candidates, we define a weighted frequency $f(m)$

$$f(m) = \sum_{i=1}^{n} e(i) \cdot s_i \ . \tag{1}$$

Where:

$$e(i) = \begin{cases} 1 & \text{if } m \in M_i \\ 0 & \text{otherwise} \ . \end{cases} \tag{2}$$

- Inverse document frequency $idf(m)$ over the training corpus:

$$idf(m) = \log(\frac{N}{1+C}) \tag{3}$$

- Final score is:

$$score(m) = f(m) \cdot idf(m) \tag{4}$$

- Lower threshold of 1.5, return the highest scored MeSH descriptors (max 12).

Assigning Descriptors Part 2

- For any MeSH header m in our set of candidates, we define a weighted frequency $f(m)$

$$f(m) = \sum_{i=1}^{n} e(i) \cdot s_i \ . \tag{1}$$

Where:

$$e(i) = \begin{cases} 1 & \text{if } m \in M_i \\ 0 & \text{otherwise} \ . \end{cases} \tag{2}$$

- Inverse document frequency $idf(m)$ over the training corpus:

$$idf(m) = \log(\frac{N}{1+C}) \tag{3}$$

- Final score is:

$$score(m) = f(m) \cdot idf(m) \tag{4}$$

- Lower threshold of 1.5, return the highest scored MeSH descriptors (max 12).

Assigning Descriptors Part 2

- For any MeSH header m in our set of candidates, we define a weighted frequency $f(m)$

$$f(m) = \sum_{i=1}^{n} e(i) \cdot s_i \ . \tag{1}$$

Where:

$$e(i) = \begin{cases} 1 & \text{if } m \in M_i \\ 0 & \text{otherwise} \ . \end{cases} \tag{2}$$

- Inverse document frequency $idf(m)$ over the training corpus:

$$idf(m) = \log(\frac{N}{1+C}) \tag{3}$$

- Final score is:

$$score(m) = f(m) \cdot idf(m) \tag{4}$$

- Lower threshold of 1.5, return the highest scored MeSH descriptors (max 12).

Outline

Results

Table : Flat Measures

Batch	System	Micro-P	Micro-R	Micro-F
3: Wk 4	Baseline	0.2466	**0.2942**	**0.2683**
3: Wk 4	mesh_lsi	**0.2815**	0.2370	0.2573
3: Wk 5	Baseline	0.2315	**0.3088**	**0.2646**
3: Wk 5	mesh_lsi	**0.2688**	0.2423	0.2549

3

Table : Hierarchical Measures (Lowest Common Ancestor)

Batch	System	LCA-P	LCA-R	LCA-F
3: Wk 4	Baseline	**0.3271**	**0.3207**	**0.3107**
3: Wk 4	mesh_lsi	0.3230	0.2699	0.2844
3: Wk 5	Baseline	0.3061	**0.3345**	**0.3059**
3: Wk 5	mesh_lsi	**0.3177**	0.2782	0.2874

[3] Baseline is the BioASQ baseline MTI and MTI First Line Index

Example

C-Reactive Protein Haplotype Predicts Serum C-Reactive Protein Levels But Not Cardiovascular Disease Risk in a Dialysis Cohort

Lin Zhang, MD, PhD, W.H. Linda Kao, PhD, MHS, Yvette Berthier-Schaad, PhD, Laura Plantinga, ScM, Nancy Fink, MPH, Michael W. Smith, PhD, and Josef Coresh, MD, PhD

Background: C-Reactive protein (*CRP*) gene variation has been associated with serum CRP levels in the general population. We examined the associations of *CRP* gene variation with longitudinal CRP measurements and incident cardiovascular disease (CVD) risk in a cohort of 504 white and 244 African-American incident dialysis patients.

Methods: Seven tagging single-nucleotide polymorphisms in the *CRP* gene were selected by using the Carlson method ($r^2 > 0.65$). High-sensitivity CRP was measured every 6 months (mean, 4.6 months). Haplo.glm was used to determine the association of haplotypes with serum CRP levels and CVD risk. Global tests from Haplo.score were conducted to determine statistical significance.

Results: Compared with the most common haplotype, 1 haplotype was associated with a 52% lower CRP level at baseline among African Americans (ratio, 0.48; 95% confidence interval [CI], 0.28 to 0.82; global *P*-value = 0.0005). Furthermore, this haplotype was associated significantly with lower serum CRP levels during 36 months of follow-up. Among whites, this haplotype was associated with an 18% (ratio, 0.82; 95% CI, 0.56 to 1.22; n = 6 carriers) lower CRP level compared with the most common haplotype with a frequency of 1% (global *P*-value = 0.048). No association was detected between *CRP* gene variation and CVD risk in either whites or African Americans.

Conclusion: Compared with the most common haplotype of the *CRP* gene, 1 haplotype predicts a lower serum CRP level over time, but no association exists between haplotype of *CRP* gene and incident CVD in this incident dialysis population. Serum CRP level might be a biomarker, rather than a causal factor, in CVD development. *CRP* variation may lead to susceptibility to inflammation, but not risk for CVD; however, replication in multiple settings is necessary.

Am J Kidney Dis 49:118-126.

INDEX WORDS: C-Reactive protein (*CRP*) gene; serum C-reactive protein (CRP) level; haplotype; cardiovascular disease (CVD); end-stage renal disease (ESRD).

Elevated serum C-reactive protein (CRP) level is associated significantly with risk for cardiovascular disease (CVD) in the general population[1,2] and the dialysis population,[3,4] who are at high risk for inflammation and CVD.[5-7] However, it is unclear whether the association between CRP level and increased CVD risk is due to reverse causality or residual confounding, which would make CRP level a marker rather than a causal risk factor, for increased CVD risk. Genetic association studies may help address this question.[8] If high serum CRP levels were a

From the Department of Epidemiology and Welch Center for Prevention, Epidemiology and Clinical Research, Johns Hopkins Bloomberg School of Public Health; Department of Medicine, Johns Hopkins School of Medicine, Baltimore; and Laboratory of Genomic Diversity and Basic Research Program, SAIC-Frederick, National Cancer Institute–Frederick, MD.

Received May 30, 2006; accepted in revised form October 10, 2006.

project has been funded in whole or part with federal funds from the National Cancer Institute, National Institutes of Health (NIH), under contract N01-CO-12400. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the US Government. This research was supported by the Intramural Research

Example, Part 2

Actual: *C-Reactive Protein, Cardiovascular Diseases, Female, Haplotypes, Humans, Male, Middle Aged, Renal Dialysis, Risk Factors*

Predicted: *Aged, Ankle Brachial Index, Biological Markers, C-Reactive Protein, Cardiovascular Diseases, Cohort Studies, Cross-Sectional Studies, Female, Logistic Models, Middle Aged, Predictive Value of Tests, Risk Factors*

Example Part 3

For this example, 147 candidates terms were considered, including all of the manually applied MeSH terms.

Table : Example Candidates and Scores for A Sample Abstract

MeSH Descriptor	Score
C-Reactive Protein	9.008
Biological Markers	5.399
Risk Factors	4.959
Cross-Sectional Studies	4.539
Logistic Models	3.513
Cardiovascular Diseases	3.322
Predictive Value of Tests	3.267
Aged	3.265
Cohort Studies	3.117
Middle Aged	2.942
Ankle Brachial Index	2.814
Female	2.558
Venous Thromboembolism	2.447
Male	2.391
...	...
Humans	1.382
...	...
Renal Dialysis	0.878
...	...
Haplotypes	0.8322
...	...

Outline

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’
- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’
- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Summary

- Results are encouraging. Seems to be a viable approach to applying semantic tags.
- There are a number of avenues that need to be explored before this can move beyond ‘proof of concept’

- Future Work
  - Possible special case handling of check tags such as “Human”
  - Improvements to the LSI
    - Better stopwords: Consider ignoring numbers and section headers.
    - Better normalization: Stemming/Lemmatization. Acronym normalization.
  - Tune variables
    - Number of LSI topics
    - Number of similar documents considered.
    - Modify or remove hard ceiling of 12 on number of assigned MeSH terms.

Questions?

adamjo@ohsu.edu