A Realm-based Question Answering System using Probabilistic Modeling

Master's Thesis

Clint P. George

Department of Computer and Information Science & Engineering
University of Florida

October 26, 2010

Outline

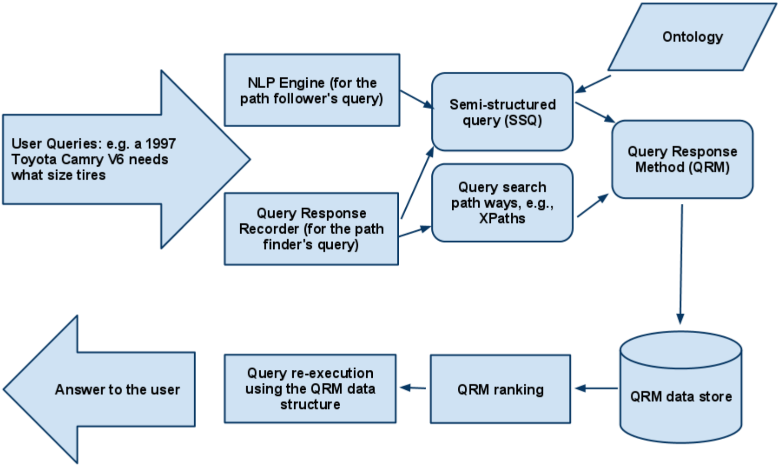
The question answering problem

- Conventional search engines are mainly based on `key-word` based search
- This makes it difficult to answer a question, which is usually in a natural language format
    - i.e. we may have to follow *several links* (pathways) to reach a web page providing an answer
- Need for an automatic system that can solve user queries

Morpheus's strategy for question answering

- A realm-based approach to answer a user query
  - e.g. automotive
- Find an answer to a query by re-visiting relevance-ranked prior search pathways
- Follow a hierarchical strategy
  - Tag query terms using classes from an ontology, track the search pathways, and store them
  - Automatically parse new queries, assign classes and a realm to terms
  - Find similar stored queries and re-run the stored pathways to present results to the user

Morpheus architecture



This thesis's contributions

- A ranking algorithm for the prior search queries
  - The class divergence quasi-metric
  - The SSQ ranking algorithm
- Tools that tackle document categorization and ontology learning problems
  - Document modeling
  - Topic extraction from the web
  - Dimensionality reduction

Ontology

An ontology formally models real world concepts and their relationships.

- *concepts* are represented by ontological *classes*, e.g. automobile
- *properties* and *property restrictions* represent concepts' *relationships and attributes*, e.g. *hasSize* property for the class *Tire*
- An ontological class can have multiple *super classes* and *sub-classes*
- Morpheus uses the ontological classes to tag *query terms*

Class corpus and term tagging

- Every leaf node (*class*) of the ontology is associated with a corpus of terms or words
- Terms's frequencies are used for calculating the probability of term given a class, which is
  - used to automatically tag classes to terms
  - also used to automatically determine a realm for a query
- Query matching is based on a realm of interest e.g. *automotive*

Terms	1997	Toyota	Camry	V6	tire size
Input	year	manufacturer	model	engine	
Output					tire_size

Table: A semi-structured query and the tagged classes

Class divergence - algorithm

To find the similarity between a path follower's SSQ, a *candidate SSQ*, and a path finder's SSQ, a *qualified SSQ*, we aggregate the similarity measures of their assigned *classes*

- Let S be the source class and T be the target class. $S \prec T$ represents the reflexive transitive closure of the superclass relation.
- $d(P, Q)$ represents the hop distance in the directed ontology inheritance graph from P to Q.
- Let C be a common ancestor class of S and T which minimizes $d(S, C) + d(T, C)$

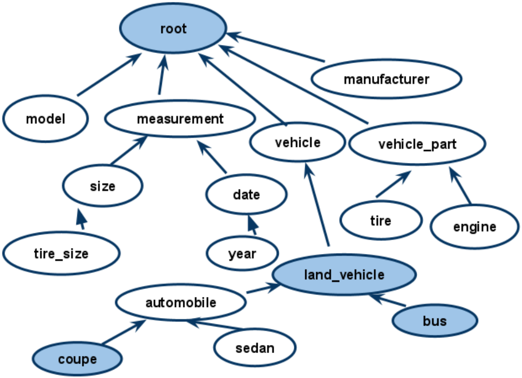
Class divergence - algorithm

The class divergence between S and T is defined as:

$$cd(S,T) = \begin{cases} 0 & S \equiv T \\ d(S,T)/(3h) & S \prec T \\ 1 & T \prec S \\ (d(S, root) + d(S,C) + d(T,C))/(3h) & otherwise \end{cases}$$

- h represents the height of the ontology tree
- $cd(S,T)$ is in the range of zero (represents identical classes) to one (represents incompatible classes)

Class divergence - Example



E.g. tree height $h = 4$, $d(bus, root) = 3$, $d(bus, land\_vehicle) = 1$, $d(coupe, land\_vehicle) = 2$, and $cd(bus, coupe) = \frac{3+1+2}{3*4}$

SSQ matching and query ranking

To find the relevance between a candidate SSQ and a qualified SSQ

- An SSQ contains terms, tagged ontological classes, and a realm

Methods

- Calculate the similarity between the SSQs based on the class divergence values of the assigned classes
- Order the QRMs in the store by increasing divergence
  - The order provides ranking for the results to the path follower's query

Query results

Suppose a path follower enters - *"What is the tire size for a 1997 Toyota Camry V6?"*

WH-question Term	what
Asking For	tire size
n-grams	1997, 1997 Toyota, 1997 Toyota Camry, Toyota, Toyota Camry, Toyota Camry V6, Camry, Camry V6, V6

Query	Tagged Classes	cd
What is the tire size for a 1998 Toyota Sienna XLE Van?	manufacturer, model, year, tire_size[1]	0.000
Where can I buy an engine for a Toyota Camry V6?	engine, manufacturer, model, vehicle_part[1]	0.216

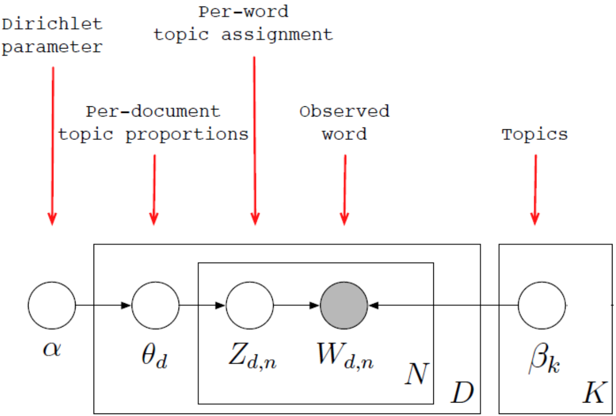
Document modeling

One approach to build an ontology and the class corpora from web documents is to categorize the web documents and learn taxonomy from them.

- Latent Dirichlet Allocation is a topic model that can extract hidden topic structure from text
- This thesis uses LDA for
  - Topic extraction from the web documents
  - Dimensionality reduction
  - Building classifiers based on the LDA outputs
  - Inferring topic mixture for the newly encountered documents from the learned LDA model, without model retraining

Latent Dirichlet Allocation - overview

- Each document is a random mixture of corpus-wide topics
- Each word is drawn from one of those topics



Feature extraction

- Building the training set and test set for the machine learning model
  - Wikipedia pages are used
  - Pages are identified using the Wikipedia category hierarchy
- Tokenization of the text
- Standardization of tokens by
  - Stemming - removing unnecessary grammatical markings (e.g. walking $\rightarrow$ walk).
  - Lemmatization - representing a set of terms by a common term Lemma (e.g. am, are, is $\rightarrow$ be; see, saw $\rightarrow$ see or saw, based on context)
  - Removing stop-words, e.g., I, you, the, a, an, etc
- Features are the terms' frequencies in a document

LDA for topic extraction - experiment

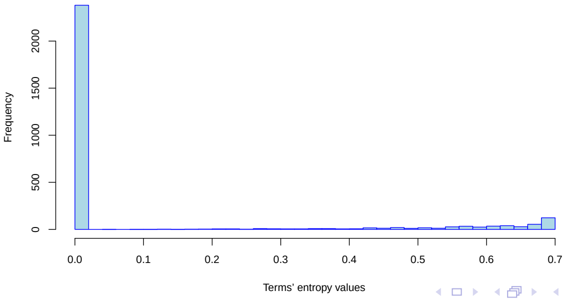
data set: Wikipedia pages from the whales (119 pages) and tires (111 pages) domains

Number	Topic 1	Topic 2
1	whale	tire
2	dolphin	tyre
3	species	wheel
4	sea	rubber
5	ship	vehicle
6	killer	tread

- LDA defines topics over all the terms in a vocabulary
- If the documents are completely different domains (e.g. whales and tires), LDA finds the topics that can be used in classifying documents

LDA for dimensionality reduction

- To find best discriminative terms in the corpus
- Based on the term-entropy values calculated on the term-topic matrix β
- High term-entropies tell us that the term is common among the corpus topics



LDA for dimensionality reduction

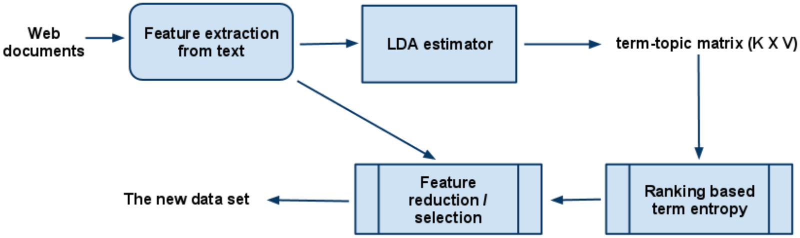


Figure: LDA based feature reduction

Document classification using document topic distance

- Find the centroid or mean $\hat{\vec{\theta}}$ of the document topic-mixtures $\vec{\theta}_d$ (for a given class)
- Calculate the minimal topic-mixture distance to the centroid

$$Hellinger(\vec{\theta}_d, \hat{\vec{\theta}}) = \sum_{k=1}^{K} (\sqrt{\vec{\theta}_d} - \sqrt{\hat{\vec{\theta}}})^2$$

$$Cosine(\vec{\theta}_d, \hat{\vec{\theta}}) = \frac{\vec{\theta}_d \, . \, \hat{\vec{\theta}}}{\left\|\vec{\theta}_d\right\| \left\|\hat{\vec{\theta}}\right\|}$$

- Use this distance measure to classify the unseen documents

Document classification using document topic distance

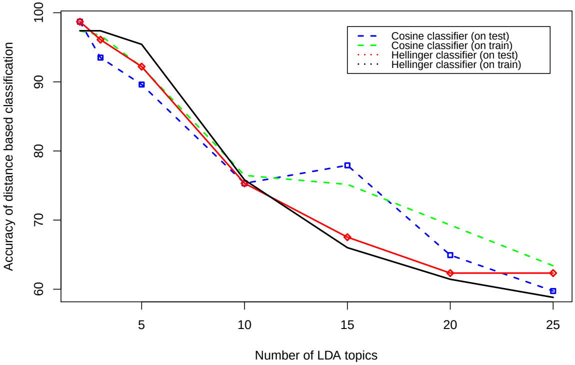


Figure: Classification accuracy of the classifiers build based on Hellinger and cosine distances with varying number of topics.

Document classification using SVM

SVM classification model based on the document topic mixture proportions from the LDA model

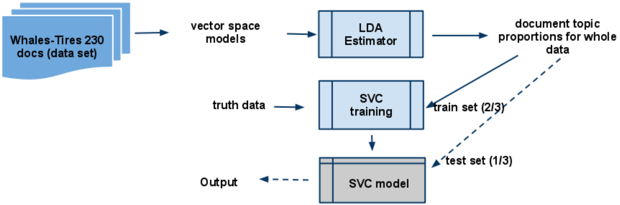


Figure: SVM classification model using the LDA document proportions

Document classification using SVM

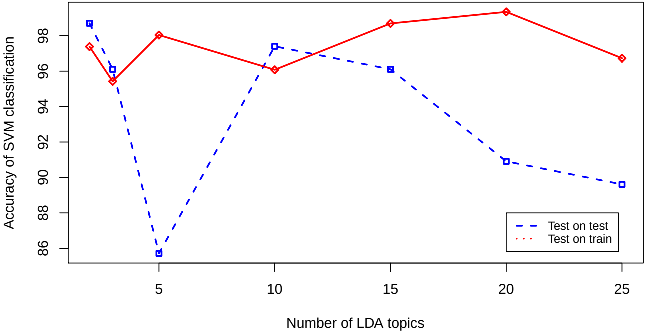


Figure: Classification accuracy of the whales-tires document topic mixtures on the varying number of topics

Topic inference using the trained LDA model

- Fitting an LDA model from documents is computationally expensive
- For the newly encountered documents, we infer document topic mixture from the learned LDA model without retraining.
  - Reduce the vocab size using the dimensionality reduction methods
  - Fit a multi-output regression model by fitting multiple regression models for each of the LDA topic proportions
  - Let t_i be the i^{th} topic row of the K X D matrix, θ, and X be D X $\acute{V}$ matrix that represents the feature vectors ($\acute{V}$) of the corpus documents. Then, the regression model is formed as:

$$t_i = f(X, w)$$

Summary

- Overview of the Morpheus architecture and question answering problem
- A query ranking algorithm based on the heterarchy of an ontology
- Document modeling and categorization techniques using LDA

- Outlook
  - Query term tagging using LDA
  - Taxonomy induction from text

Thanks

Chair	Joseph N. Wilson
Committee	Paul D. Gader Sanjay Ranka
Colleagues	Morpheus Team

For Further Reading I

Blei, David M., Ng, Andrew Y., and Jordan, Michael I.
Latent dirichlet allocation.
Journal of Machine Learning Research, 3:993–1022, 2003.

Grant, Christan, P. George, Clint, Gumbs, Joir-dan, Wilson, Joseph N., Dobbins, Peter J.
Morpheus: A Deep Web Question Answering System,
International Conference on Information Integration and Web-based Applications and Services, 2010.

SVM as a regression and classification model

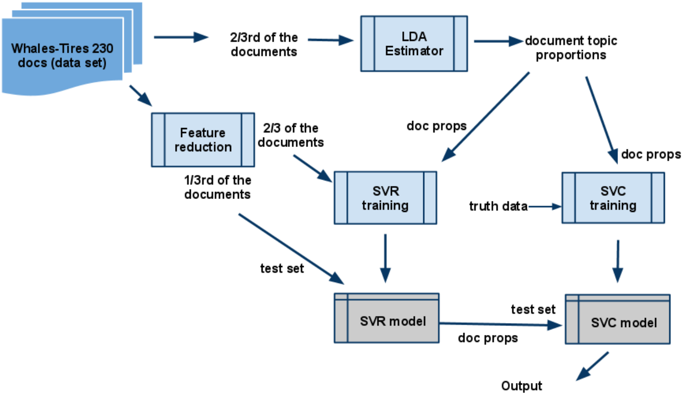


Figure: SVM as a regression and classification model with the LDA model trained on $2/3^{rd}$ of the whales and tires documents

SVM classification model

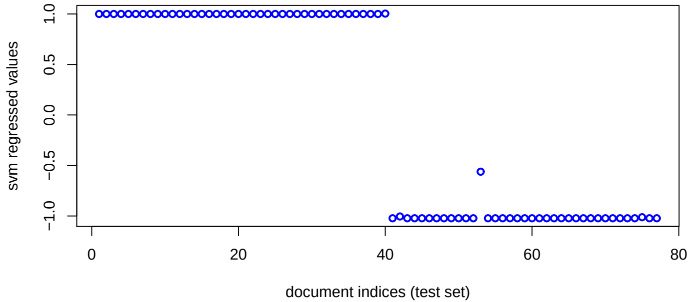


Figure: SVC regressed values of the whales-tires document topic mixtures with two topics.

LDA's topic proportions

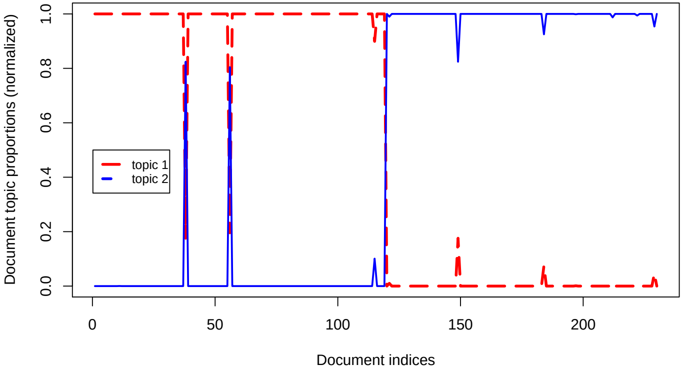


Figure: The two topic LDA's topic proportions for the whales (first 119) and tires (last 111) documents