CS 175, Project in Artificial Intelligence

Lecture 5: Projects

Padhraic Smyth
Department of Computer Science
Bren School of Information and Computer Sciences
University of California, Irvine
Announcements

• Assignment 2: Text Classification
  – Due on Friday at noon

• Project Proposals
  – Due Friday next week

• Lectures:
  – Today: discussion of projects
  – Monday: more discussion of projects and project proposals
  – Wednesday: no lecture. I will be available to meet students in my office (2 to 3:20) to answer questions about projects and proposals
# Weekly Schedule (subject to change)

<table>
<thead>
<tr>
<th>Week</th>
<th>Monday</th>
<th>Wednesday</th>
<th>Deadlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 9</td>
<td>Introduction and course outline</td>
<td>Basic concepts in automated text analysis</td>
<td></td>
</tr>
<tr>
<td>Jan 16</td>
<td>No class (university holiday)</td>
<td>Text classification</td>
<td>Assignment 1 (by Wednesday noon)</td>
</tr>
<tr>
<td>Jan 23</td>
<td>Text classification (continued)</td>
<td>Ideas for class projects</td>
<td>Assignment 2 (by Friday noon)</td>
</tr>
<tr>
<td>Jan 30</td>
<td>Discussion of project proposals</td>
<td>Office hours (no lecture)</td>
<td>Project proposal (by Friday noon)</td>
</tr>
<tr>
<td>Feb 6</td>
<td>Evaluation methods</td>
<td>Word prediction methods</td>
<td></td>
</tr>
<tr>
<td>Feb 13</td>
<td>Topic modeling algorithms</td>
<td>Office hours (no lecture)</td>
<td></td>
</tr>
<tr>
<td>Feb 20</td>
<td>No class (university holiday)</td>
<td>Office hours (no lecture)</td>
<td>Progress report (by Friday noon)</td>
</tr>
<tr>
<td>Feb 27</td>
<td>Office hours (no lecture)</td>
<td>Office hours (no lecture)</td>
<td></td>
</tr>
<tr>
<td>Mar 6</td>
<td>Project Presentations (in class)</td>
<td>Project Presentations (in class)</td>
<td></td>
</tr>
<tr>
<td>Mar 13</td>
<td>Office hours (no lecture)</td>
<td>Office hours (no lecture)</td>
<td></td>
</tr>
</tbody>
</table>
Rules for Projects

• 1 or 2 students per project
  – For 2-student projects I expect twice as much work as 1-student projects
  – 2-person teams: each will need to submit individual progress and final reports

• Use of external code is allowed and encouraged
  – Such code needs to be acknowledged in your reports

• You must write at least some functionality on your own
  – What you implement is up to you
  – You will need to submit the code at the end of the quarter
Planning a Project

• Topic: select a general type of problem you are interested in, e.g.,
  – Classification, information extraction, summarization, synthesis, parsing, etc

• Do some background reading to learn more about the topic
  – e.g., from chapters in the books I recommended
  – e.g., from links on class Web site

• Define your problem precisely

• Determine at least 1 or 2 data sets you can use for your project

• Figure out how you will evaluate your results, e.g., to compare A v B
  – Experiments: classification accuracy, precision/recall, etc
  – User Studies: human users compare results from A and B
Project-Related Reference Material for CS 175

CS 175, Winter 2017
Below are links to suggested reading organized by topic. If you are doing a project on any of these topics (or interested in potentially doing a project on these topics) then these online resources should be helpful.

Text Classification
Chapter on text classification and naive Bayes from Manning et al
Chapter on vector-based classification for text from Manning et al
Chapter on logistic regression for document classification from Jurafsky and Martin
Comprehensive survey paper on text classification algorithms by Aggarwal and Zhai (2012)
Overview of general principles in machine learning from Goodfellow et al (2016)
Tutorial paper on multi-label classification methods by de Carvalho and Freitas

Sentiment Analysis
Chapter on naive Bayes and sentiment classification from Jurafsky and Martin
Survey papers on opinion mining and sentiment analysis: by Pang and Lee (2008) and by Liu and Zhang (2012)

Sequential Classifiers
Chapter on recurrent and recursive neural networks from Goodfellow et al (2016)

Document Clustering
Chapters on flat clustering algorithms and hierarchical clustering algorithms for text documents, from Manning et al

Topic Modeling
Tutorial paper on topic modeling by Steyvers and Griffiths
David Blei’s page on topic modeling

Vector Embeddings
Chapter on dense vector representations for words from Jurafsky and Martin
Chapter on latent semantic indexing from Manning et al
## Project Deliverables and Deadlines

<table>
<thead>
<tr>
<th>Deliverable</th>
<th>Grade Percentage</th>
<th>Date Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Proposal</td>
<td>20%</td>
<td>Friday Feb 3rd</td>
</tr>
<tr>
<td>Progress Report</td>
<td>20%</td>
<td>Friday Feb 24th</td>
</tr>
<tr>
<td>Presentation/Demo</td>
<td>10%</td>
<td>March 6(^{th})/8th</td>
</tr>
<tr>
<td>Final Report</td>
<td>30%</td>
<td>Monday March 20th</td>
</tr>
</tbody>
</table>
### Examples of Types of Projects

<table>
<thead>
<tr>
<th>Type</th>
<th>Goal</th>
<th>Examples of Data Sets</th>
<th>Evaluation Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Predict sentiment (pos/neg) or stars from text</td>
<td>Twitter, Yelp, Product or Movie Reviews</td>
<td>Classification accuracy, Precision-recall, ...</td>
</tr>
<tr>
<td></td>
<td>Predict dialog acts in transcribed conversations</td>
<td>Switchboard corpus</td>
<td></td>
</tr>
<tr>
<td>Text Summarization</td>
<td>Summarize a set of documents</td>
<td>Product or Movie Reviews</td>
<td>BLEU scores, User studies</td>
</tr>
<tr>
<td>Information Extraction</td>
<td>Product or restaurant reviews</td>
<td>Product Reviews News Articles</td>
<td>Accuracy (if labeled), otherwise user studies</td>
</tr>
<tr>
<td>Text Synthesis</td>
<td>Generate new text in the style of an author</td>
<td>Articles/books/songs by different authors</td>
<td>User studies</td>
</tr>
<tr>
<td>Question Answering</td>
<td>Generate an answer to a question</td>
<td>Q&amp;A data sets for research projects</td>
<td>Accuracy, precision-recall</td>
</tr>
</tbody>
</table>
Project Proposals

• Your proposal should be 2 to 3 pages long
  – Required to use project proposal template (will be posted by this weekend)

• Project proposals will be graded like a homework assignment and receive a weight of 20% of your overall grade.

• Proposals will primarily be graded on
  – (a) clarity (is it clear what will be done in this project?) and
  – (b) completeness (does the proposal address all of the important aspects of the proposed project?)

• Note: if a project is too simple (or too complex!), or missing important details, it may be returned to you and a revision requested.

(Assignment for Project Proposals will be available on Webpage by end of day Friday)
Contents of Project Proposal

1. Project Definition (1 to 2 sentences)

2. Problem Description and Background

3. Data Sets

4. Proposed Technical Approach

5. Experiments and Evaluation

6. Software

7. Milestones

8. (For Teams) Individual Student Responsibilities
Project Teams

• Students can work individually or in 2-person project teams

• Important: proposal needs to clearly identify what will be worked on by each individual student.

• Submit only one project proposal per team (include both team member names and IDs on the proposal)

• Both team members will get the same grade for the proposal.
Software Development

- You will likely use both
  - code that your team writes for the project (required)
  - publicly-available code, e.g., from scikit-learn or NLTK (optional but encouraged)

- A typical pipeline will contain a mix of code
  - If you did not write the code, you need to acknowledge the source

- Part of your project could be implementing an algorithm or technique that already exists as publicly-available code, e.g.,
  - A parser
  - A neural network classifier
General Criteria for Grading of Projects

• Technical competence
  – Were algorithms and methods used correctly and appropriately?
  – Did students understand the methods they were using?
  – Were systematic experiments conducted and results interpreted?

• Effort
  – How much work was done (e.g., coding, experiments, background reading, etc)
  – Note that effort alone is necessary but not sufficient for a high grade!

• Creativity and insight:
  – Did you demonstrate creativity in your project?
  – What did insights did you gain from the project?

• Writing and communication:
  – are you able to explain your work clearly?
Possible Project Topics
Question:

How can we plan out the details of our project when we haven’t yet done any work on this topic?

Your project proposal should be your best estimate at this time of what your project will be about - with the realization that there may need to be changes and adjustments along the way.

- For example, you might find that the initial data set you chose is not really suitable and need to change data sets.
- Or that the algorithm you planned to investigate is not really suitable to the task and need to change to another algorithm.
- Or that the problem you are addressing is too difficult and that you need to pick a simpler one.
Ideas for Finding Project Topics

- Refer to past lecture slides and the rest of these slides

- Browse and read articles in *Links to Tutorial Articles*... on the class Web page

- Browse paper titles and abstracts from research conferences, use search engines (e.g., Google Scholar) to search on specific topics

- Post privately or publicly on Piazza to get some general initial feedback, e.g., “would this idea be sufficient for a project?”

- Come to office hours next week (Mon, Tues, Wed)
Project-Related Reference Material for CS 175

CS 175, Winter 2017
Below are links to suggested reading organized by topic. If you are doing a project on any of these topics (or interested in potentially doing a project on these topics) then these online resources should be helpful.

Text Classification
Chapter on text classification and naive Bayes from Manning et al
Chapter on vector-based classification for text from Manning et al
Chapter on logistic regression for document classification from Jurafsky and Martin
Comprehensive survey paper on text classification algorithms by Aggarwal and Zhai (2012)
Overview of general principles in machine learning from Goodfellow et al (2016)
Tutorial paper on multi-label classification methods by de Carvalho and Freitas

Sentiment Analysis
Chapter on naive Bayes and sentiment classification from Jurafsky and Martin
Survey papers on opinion mining and sentiment analysis: by Pang and Lee (2008) and by Liu and Zhang (2012)

Sequential Classifiers
Chapter on recurrent and recursive neural networks from Goodfellow et al (2016)

Document Clustering
Chapters on flat clustering algorithms and hierarchical clustering algorithms for text documents, from Manning et al

Topic Modeling
Tutorial paper on topic modeling by Steyvers and Griffiths
David Blei’s page on topic modeling

Vector Embeddings
Chapter on dense vector representations for words from Jurafsky and Martin
Chapter on latent semantic indexing from Manning et al
Different Types of Projects

• Method-focused
  – Comparing different algorithms (e.g., for document classification)
    • E.g., compare naïve Bayes, logistic regression, and neural networks
    • Evaluate on multiple different data sets, extensive experiments
  – Compare different text representation methods
    • Different methods for extracting n-grams
    • The effect of stop words
  – Test the sensitivity of classifiers to (e.g.,)
    • Vocabulary size, document length, number of documents, etc

• Application-focused
  – Focus on a particular problem, evaluate different methods
    • Sentiment classification for Twitter or other social media
    • Classifying utterances in conversation into dialog acts
    • Spam detection
  – Typically trying to see how well an automated system can do on a task
Possible Topics for Projects

• **Document classification**
  – Bag-of-words: sentiment analysis, spam detection, hierarchical classifiers
  – Sequential classifiers: e.g., classify a sentence using word sequence

• **Unsupervised learning:**
  – Techniques: clustering, topic modeling, word embeddings
  – Develop a tool to help explore and visualize large text corpora

• **Text summarization and information extraction**
  – Extracting specific information from product/restaurant/movie reviews
  – Summarizing a large set of reviews or documents

• **Text generation/simulation/synthesis**

• **Other ideas**
  – Question-answering systems
  – Identifying authors from the text they write
Document Classification
Example: Predicting Ratings given Text of Review Data

- Investigate how well star ratings can be predicted as a function of the number of words in a document
  - i.e., take only the first $K$ words in a review
    - Vary $K$, e.g., $K = 1, 2, 5, 10, 20$
  - Build classifiers only using these words (ignore the rest)
  - What is the best accuracy you can get as a function of $K$
  - Could also try other variations (e.g., $K$ randomly selected words)

- Can neural networks improve significantly on logistic regression for rating prediction?
  - We saw with Dimitris’ code that we can get 94% to 95% accuracy on Yelp
  - Could a more complex neural network do better?
  - Perhaps make the problem harder and predict all 5 categories
    - Or predict 3 classes: \{1,2\} or 3 or \{4,5\}
Chapter 13

A SURVEY OF OPINION MINING AND SENTIMENT ANALYSIS

Bing Liu
University of Illinois at Chicago
Chicago, IL
liub@cs.uic.edu

Lei Zhang
University of Illinois at Chicago
Chicago, IL
lzhang32@gmail.com

Abstract  Sentiment analysis or opinion mining is the computational study of people’s opinions, appraisals, attitudes, and emotions toward entities, individuals, issues, events, topics and their attributes. The task is technically challenging and practically very useful. For example, businesses always want to find public or consumer opinions about their products and services. Potential customers also want to know the opinions of existing users before they use a service or purchase a product.

With the explosive growth of social media (i.e., reviews, forum discussions, blogs and social networks) on the Web, individuals and organizations are increasingly using public opinions in these media for
Consumer Confidence and Twitter Sentiment

## Named Entity Recognition using Classification

<table>
<thead>
<tr>
<th>Type</th>
<th>Tag</th>
<th>Sample Categories</th>
<th>Example sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>PER</td>
<td>people, characters</td>
<td>Turing is a giant of computer science.</td>
</tr>
<tr>
<td>Organization</td>
<td>ORG</td>
<td>companies, sports teams</td>
<td>The IPCC warned about the cyclone.</td>
</tr>
<tr>
<td>Location</td>
<td>LOC</td>
<td>regions, mountains, seas</td>
<td>The Mt. Sanitas loop is in Sunshine Canyon.</td>
</tr>
<tr>
<td>Geo-Political</td>
<td>GPE</td>
<td>countries, states, provinces</td>
<td>Palo Alto is raising the fees for parking.</td>
</tr>
<tr>
<td>Entity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facility</td>
<td>FAC</td>
<td>bridges, buildings, airports</td>
<td>Consider the Tappan Zee Bridge.</td>
</tr>
<tr>
<td>Vehicles</td>
<td>VEH</td>
<td>planes, trains, automobiles</td>
<td>It was a classic Ford Falcon.</td>
</tr>
</tbody>
</table>

**Figure 21.1** A list of generic named entity types with the kinds of entities they refer to.
Possible Topic: Document Classification, Sequential Models

- Example: classifying words within a sentence into one of \{person, organization, place, other\}

- Algorithms that use surrounding context words are more accurate
  - E.g., “Jobs said that Apple will ....” v. “Jobs are becoming more plentiful...”
  - Parts of speech of surrounding words can be quite important

- Classifiers need to use not just the word itself but also nearby words and their properties
  - Window-based approaches: train classifiers on words and POS within plus/minus K words (e.g., K = 3)
  - Sequential models, e.g., recurrent neural networks
identity of $w_i$
identity of neighboring words
part of speech of $w_i$
part of speech of neighboring words
base-phrase syntactic chunk label of $w_i$ and neighboring words
presence of $w_i$ in a gazetteer
$w_i$ contains a particular prefix (from all prefixes of length $\leq 4$)
$w_i$ contains a particular suffix (from all suffixes of length $\leq 4$)
$w_i$ is all upper case
word shape of $w_i$
word shape of neighboring words
short word shape of $w_i$
short word shape of neighboring words
presence of hyphen

**Figure 21.5** Features commonly used in training named entity recognition systems.
Figure 21.7  Named entity recognition as sequence labeling. The features available to the classifier during training and classification are those in the boxed area.
Unsupervised Text Analysis Algorithms
Possible Topic: Unsupervised Document Analysis

- Algorithms that work with unlabeled/unsupervised data
  - Useful as tools to help humans explore, understand, visualize large data sets
  - Potentially useful as preprocessing for supervised algorithms

- Document Clustering, e.g., of count or tfidf vector
  - Flat clustering, e.g., k-means
  - Hierarchical clustering -> produces trees of clusters
Possible Topic: Unsupervised Document Analysis

- Algorithms that work with unlabeled/unsupervised data
  - Useful as tools to help humans explore, understand, visualize large data sets
  - Potentially useful as preprocessing for supervised algorithms

- Document Clustering, e.g., of count or tfidf vector
  - Flat clustering, e.g., k-means
  - Hierarchical clustering -> produces trees of clusters

- Topic Modeling
  - Statistical model for discovering topics: documents = mixtures of topics

- Semantic Vector Representations
  - Latent semantic indexing (LSI): matrix algebra to find vectors for words and docs
  - Word embedding: neural network ideas for finding word-vectors
  - Resulting word and document vectors capture semantic similarity
Possible Topic: Unsupervised Document Analysis

- Algorithms that work with unlabeled/unsupervised data
  - Useful as tools to help humans explore, understand, visualize large data sets
  - Potentially useful as preprocessing for supervised algorithms

- Document Clustering, e.g., of count or tfidf vector
  - Flat clustering, e.g., k-means
  - Hierarchical clustering -> produces trees of clusters

- Topic Modeling
  - Statistical model for discovering topics: documents = mixtures of topics

- Semantic Vector Representations
  - Latent semantic indexing (LSI): matrix algebra to find vectors for words and docs
  - Word embedding: neural network ideas for finding word-vectors
  - Resulting word and document vectors capture semantic similarity

- Issue: evaluation of unsupervised methods can be challenging
Example of latent semantic indexing for movies, Figure from Koren, Bell, Volinksy, IEEE Computer, 2009
Word Embeddings

**Input layer**
1-hot input vector

**Projection layer**
embedding for $w_t$

**Output layer**
probabilities of context words

\[ W \] $|V| \times d$

\[ C \] $d \times |V|$

\[ W_{t+1} \]

\[ x_1 \]
\[ x_2 \]
\[ \vdots \]
\[ x_j \]
\[ \vdots \]
\[ x_{|V|} \]

\[ y_1 \]
\[ y_2 \]
\[ \vdots \]
\[ y_k \]
\[ \vdots \]
\[ y_{|V|} \]
From: https://blog.kaggle.com/2016/05/18/home-depot-product-search-relevance-winners-interview-1st-place-alex-andreas-nurlan/
Examples: what Words are Close in the Embedded Space?

<table>
<thead>
<tr>
<th>FRANCE</th>
<th>JESUS</th>
<th>XBOX</th>
<th>REDDISH</th>
<th>SCRATCHED</th>
<th>MEGABITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUSTRIA</td>
<td>GOD</td>
<td>AMIGA</td>
<td>GREENISH</td>
<td>NAILED</td>
<td>OCTETS</td>
</tr>
<tr>
<td>BELGIUM</td>
<td>SATI</td>
<td>PLAYSTATION</td>
<td>BLUISH</td>
<td>SMASHED</td>
<td>MB/S</td>
</tr>
<tr>
<td>GERMANY</td>
<td>CHRIST</td>
<td>MSX</td>
<td>PINKISH</td>
<td>PUNCHED</td>
<td>BIT/S</td>
</tr>
<tr>
<td>ITALY</td>
<td>SATAN</td>
<td>IPOD</td>
<td>PURPLISH</td>
<td>POPPED</td>
<td>BAUD</td>
</tr>
<tr>
<td>GREECE</td>
<td>KALI</td>
<td>SEGA</td>
<td>BROWNISH</td>
<td>CRIMPED</td>
<td>CARATS</td>
</tr>
<tr>
<td>SWEDEN</td>
<td>INDRA</td>
<td>PSNUMBER</td>
<td>GREYISH</td>
<td>SCRAPEDE</td>
<td>KBIT/S</td>
</tr>
<tr>
<td>NORWAY</td>
<td>VISHNU</td>
<td>HD</td>
<td>GRAYISH</td>
<td>SCREWED</td>
<td>MEGAHERTZ</td>
</tr>
<tr>
<td>EUROPE</td>
<td>ANANDA</td>
<td>DREAMCAST</td>
<td>WHITISH</td>
<td>SECTIONED</td>
<td>MEGAPIXELS</td>
</tr>
<tr>
<td>HUNGARY</td>
<td>PARVATI</td>
<td>GEFORCE</td>
<td>SILVERY</td>
<td>SLASHED</td>
<td>GBIT/S</td>
</tr>
<tr>
<td>SWITZERLAND</td>
<td>GRACE</td>
<td>CAPCOM</td>
<td>YELLOWISH</td>
<td>RIPED</td>
<td>AMPERES</td>
</tr>
</tbody>
</table>

From Collobert et al, 2011
Vector Embeddings can capture Semantic Relations

\[ \text{vector('king')} - \text{vector('man')} + \text{vector('woman')} \approx \text{vector('queen')} \]

\[ \text{vector('Paris')} - \text{vector('France')} + \text{vector('Italy')} \approx \text{vector('Rome')} \]
Information Extraction, Text Summarization
Idea: Discovering what Customers Like or Dislike

- A useful system (for readers of reviews) would be one that can also summarize what aspects of a restaurant (or product) the users like or dislike

- An example of how this might work
  - For each restaurant find all the negative reviews (1 star or 2 stars for example)
  - Across all these reviews for this restaurant
    - Search for terms (e.g., nouns and noun phrases) that are occurring more often in these negative reviews than they are in the positive reviews
    - And/or search for specific terms that are associated with known aspects of a restaurant, e.g., related to food, service, price, atmosphere, parking, etc
    - Analyze the terms and their frequencies to produce an automatic summary of what terms appear to explain why this restaurant is getting negative reviews
      - E.g., could produce scores (on some scale) for different aspects

- Note: to do this well may require looking at adjectives and negations, e.g.,
  - “the service was terrific but the food was awful”
  - “the atmosphere was not good”
Possible Topic: Text Summarization

• Automatically generate a short (2 to 5 sentence) summary of a document
  – Useful in a wide variety of applications
  – Can also be applied to a set of documents (e.g., a set of reviews)

• Key ideas
  – Find important or significant words in a document
  – Score sentences in terms of the importance of the words in them
  – Use the high-scoring sentences to generate a summary
    • Can use the original sentences in the summary
    • Or can use phrases from the sentences and generate new sentences
  – The summary should be readable, short, have good coverage of the document

• Evaluation
  – Extrinsic, e.g., human evaluation
  – Intrinsic, e.g., compute some notion of “coverage”
    • The ROUGE evaluation metric is widely used
Automatic Summarization

By Ani Nenkova and Kathleen McKeown

Contents

1 Introduction 104

1.1 Types of Summaries 104
1.2 How do Summarization Systems Work? 107
1.3 Evaluation Issues 114
1.4 Where Does Summarization Help? 115
1.5 Article Overview 117

2 Sentence Extraction: Determining Importance 120

2.1 Unsupervised Data-driven Methods 121
2.2 Machine Learning for Summarization 131
2.3 Sentence Selection vs. Summary Selection 134
2.4 Sentence Selection for Query-focused Summarization 136
2.5 Discussion 141
**Topic signatures** As his lawyers in London tried to quash a Spanish arrest warrant for Gen. Augusto Pinochet, the former Chilean Dictator, efforts began in Geneva and Paris to have him extradited. Britain has defended its arrest of Gen. Augusto Pinochet, with one lawmaker saying that Chile’s claim that the former Chilean Dictator has diplomatic immunity is ridiculous. Margaret Thatcher entertained former Chilean Dictator Gen. Augusto Pinochet at her home two weeks before he was arrested in his bed in a London hospital, the ex-prime minister’s office said Tuesday, amid growing diplomatic and domestic controversy over the move.

**Human abstract** Former Chilean dictator Augusto Pinochet has been arrested in London at the request of the Spanish government. Pinochet, in London for back surgery, was arrested in his hospital room. Spain is seeking extradition of Pinochet from London to Spain to face charges of murder in the deaths of Spanish citizens in Chile under Pinochet’s rule in the 1970s and 1980s. The arrest raised confusion in the international community as the legality of the move is debated. Pinochet supporters say that Pinochet’s arrest is illegal, claiming he has diplomatic immunity. The final outcome of the extradition request lies with the Spanish courts.
Question-Answering Systems
WikiQA: A Challenge Dataset for Open-Domain Question Answering

Yi Yang, Wen-tau Yih, and Christopher Meek
21 September 2015

Abstract

We describe the WikiQA dataset, a new publicly available set of question and sentence pairs collected and annotated for research on open-domain question answering. Most previous work on answer sentence selection focuses on a dataset created using the TREC-QA data, which includes editor-generated questions and candidate answer sentences selected by matching content words in the question. WikiQA is constructed using a more natural process and is more an order of magnitude larger than the previous dataset. In addition, the WikiQA dataset also includes questions for which there are no correct sentences, enabling researchers to work on answer triggering, a critical component in any QA system. We compare several systems on the task of answer sentence selection on both datasets and also describe the performance of a system on the problem of answer triggering using the WikiQA dataset.

Details

- **Publication type**: Inproceedings
- **Published in**: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing
- **Publisher**: ACL – Association for Computational Linguistics

Publication files

-YangYihMeek_EMNLP-15_WikiQA.pdf
-data-code_link.txt
-EMNLP-15-WikiQA_Final_Day.pdf
-bibtex.bib

By the same authors

- Embedding Entities and Relations for Learning and Inference in Knowledge Bases
- Online Discriminative Spam Filter Training
- Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base
DeepMind Q&A Dataset

Hermann et al. (2015) created two awesome datasets using news articles for Q&A research. Each dataset contains many documents (90k and 197k each), and each document companies on average 4 questions approximately. Each question is a sentence with one missing word/phrase which can be found from the accompanying document/context.

The original authors kindly released the scripts and accompanying documentation to generate the datasets (see here). Unfortunately due to instability of WaybackMachine, it is often cumbersome to generate the datasets from scratch using the provided scripts. Furthermore, in certain parts of the world, it turned out to be far from being straight-forward to access the WaybackMachine.

I am making the generated datasets available here. This will hopefully make the datasets used by a wider audience and lead to faster progress in Q&A research.


CNN
- Questions: here
- Stories: here

This dataset contains the documents and accompanying questions from the news articles of CNN. There are approximately 90k documents and 380k questions. I am making available 'questions/', which should be sufficient to reproduce the setting from the original paper, and 'stories/', which can be useful for other uses of this dataset.

Daily Mail
- Questions: here
- Stories: here

This dataset contains the documents and accompanying questions from the news articles of Daily Mail. There are approximately 197k documents and 879k questions. I am making available 'questions/', which should be sufficient to reproduce the setting from the original paper, and 'stories/', which can be useful for other uses of this dataset.
About the Dataset

With massive volumes of written text being produced every second, how do we make sure that we have the most recent and relevant information available to us? Maluuba is tackling this problem by building AI systems that can read and comprehend large volumes of complex text in real-time.

The purpose of Maluuba's NewsQA dataset is to help the research community build algorithms that are capable of answering questions requiring human-level comprehension and reasoning skills.

Leveraging CNN articles from the DeepMind Q&A Dataset, we prepared a crowd-sourced machine reading comprehension dataset of 120K Q&A pairs.

- Documents are CNN news articles.
- Questions are written by human users in natural language.
- Answers may be multiword passages of the source text.
- Questions may be unanswerable.
Examples of Public Data Sets
Data Sets

- For most of your projects you will want to use at least one large data set (or “corpus”) for your project

- There are many real-world publicly-available data sets available for research purposes that you can use, e.g,
  - Yelp Dataset Challenge
  - 20 newsgroups data
  - ....and many more
  - See links to Data Sets on Class Web page: http://www.ics.uci.edu/~smyth/courses/cs175/reading/text_data_sets.html

- You can also gather your own data
  - E.g., by crawling Web sites or using APIs, e.g., the Twitter API
Examples of Data Sets for Text Analysis

CS 175, Winter 2017

The links below point to just a few of the many data sets for text analysis that you can find on the Web, and should help you in terms of finding data sets to work on for your projects.

Data Sets with Classification Labels or Ratings

Yelp Data Set Challenge (2.2M reviews of businesses from over 500k users in 10 cities)
(and here's a pointer to work from our own group at UCI that recently won the Round 5 Challenge)
Kaggle Data Sets. Contains multiple data sets with text content. Kaggle is a company that hosts data mining/prediction competitions
Movie review data for sentiment analysis, from Pang and Lee, Cornell
Product review data from Johns Hopkins University (goal is to predict ratings on scale of 1 to 5)
A variety of different text data sets from the UCI Machine Learning Repository (many already in the "bag of words" format)
Data Sets on "learning to rank" (for Web search)
All of Wikipedia (can be used to build classifiers using category labels or to provide additional information for other models such as n-gram statistics)
Various text and Web-related data sets from Yahoo! Labs (note that these data sets can also be used for unsupervised learning, such as clustering or topic modeling, by ignoring the class labels during training).
Document classification data sets (a large collection of different data sets used in text classification research)

Other Interesting Text Data Sets (often used for Clustering and other Exploratory Methods)

Enron email data set, from CMU (note that there are other "cleaner" versions available on the Web if you search...)
Python code for downloading IMDB (Internet Movie Database), with 425k titles and 1.7 million filmographies of cast and crew
A survey of data sets available for building data-driven dialogue systems
Book Summaries Corpus
Full text of US patents from 1980 to 2015, from the USPTO (US Patent and Trademark Office), hosted by Google
Very large data set of all Reddit submissions between 2006 and 2015

Data Sets used to build Language Models and Auto-complete Algorithms

Ngram data from Peter Norvig (Google), with an accompanying tutorial book chapter
Google ngrams, and Google syntactic ngrams over time, from Google books

Question-Answering Data Sets

WikiQA, a data set for "open-domain" question answering, from Microsoft Research
Question-Answering Data Sets from TREC (funding by the National Institute of Standards and Technology, NIST)
Question Answering Corpus from DeepMind (part of Google)
The Allen AI Science Challenge on Kaggle (competition ended in 2016)
The BioASQ data sets and challenge competitions on question answering for the biomedical domain

Ontologies/Structured Data (useful for Information Extraction/Annotation)

The DBpedia Data Set
Welcome to Kaggle Datasets

The best place to discover and seamlessly analyze publicly available data.

Dig In
Explore a dataset with our in-browser analytics tool, Kaggle Scripts. You can also download it in an easy to read format.

Build
Create your data science portfolio. Publish insights and code with Kaggle Scripts and it will be saved to your profile.

Connect
Engage with other data scientists. Share feedback on other Kagglers' scripts, or ask a question in a dataset's forum.

Hillary Clinton's Emails
337 Scripts - 19 Topics

UCI Iris
46 Scripts - 4 Topics

Ocean Ship Logbooks (1750+)
117 Scripts - 24 Topics
<table>
<thead>
<tr>
<th>Name</th>
<th>Data Types</th>
<th>Default Task</th>
<th>Attribute Types</th>
<th># Instances</th>
<th># Attributes</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D Road Network (North Jutland, Denmark)</td>
<td>Sequential, Text</td>
<td>Regression, Clustering</td>
<td>Real</td>
<td>434874</td>
<td>4</td>
<td>2013</td>
</tr>
<tr>
<td>Amazon Commerce reviews set</td>
<td>Multivariate, Text, Domain-Theory</td>
<td>Classification</td>
<td>Real</td>
<td>1500</td>
<td>10000</td>
<td>2011</td>
</tr>
<tr>
<td>Badges</td>
<td>Univariate, Text</td>
<td>Classification</td>
<td></td>
<td>294</td>
<td>1</td>
<td>1994</td>
</tr>
<tr>
<td>Bag of Words</td>
<td>Text</td>
<td>Clustering</td>
<td>Integer</td>
<td>8000000</td>
<td>100000</td>
<td>2008</td>
</tr>
<tr>
<td>CNAE-9</td>
<td>Multivariate, Text</td>
<td>Classification</td>
<td>Integer</td>
<td>1090</td>
<td>857</td>
<td>2012</td>
</tr>
<tr>
<td>DBWorld e-mails</td>
<td>Text</td>
<td>Classification</td>
<td></td>
<td>64</td>
<td>4702</td>
<td>2011</td>
</tr>
<tr>
<td>Dresses_Attribute_Sales</td>
<td>Text</td>
<td>Classification, Clustering</td>
<td></td>
<td>501</td>
<td>13</td>
<td>2014</td>
</tr>
<tr>
<td>Farm Ads</td>
<td>Text</td>
<td>Classification</td>
<td></td>
<td>4143</td>
<td>54877</td>
<td>2011</td>
</tr>
</tbody>
</table>
The Switchboard Dialog Act Corpus

1. Overview
2. Getting and using the corpus
   2.1 Downloads
   2.2 Python classes (preferred)
      2.2.1 Transcript objects
      2.2.2 Utterance objects
      2.2.3 CorpusReader objects
   2.3 Working directly with the CSV file (dispreferred but okay)
3. Annotations
   3.1 Dialog act annotations
   3.2 Penn Discourse Treebank 3 POS
   3.3 Penn Discourse Treebank 3 Trees
4. Exercises

1 Overview

The Switchboard Dialog Act Corpus (SwDA) extends the Switchboard-1 Telephone Speech Corpus, Release 2, with tu
the associated turn. The SwDA project was undertaken at UC Boulder in the late 1990s.

Recommended reading:

- Dialog Act Coders' Manual
- Stolcke et al. 2000

Code and data:

- swda.zip: our distribution of the data
- swda.py: Python classes for working with swda
- swda_functions.py: auxiliary functions for using swda.py
### Examples of Dialog Acts (Labels for Utterances)

<table>
<thead>
<tr>
<th>name</th>
<th>act_tag</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statement-non-opinion</td>
<td>sd</td>
<td>Me, I'm in the legal department.</td>
</tr>
<tr>
<td>Acknowledge (Backchannel)</td>
<td>b</td>
<td>Uh-huh.</td>
</tr>
<tr>
<td>Statement-opinion</td>
<td>sv</td>
<td>I think it's great</td>
</tr>
<tr>
<td>Agree/Accept</td>
<td>aa</td>
<td>That's exactly it.</td>
</tr>
<tr>
<td>Abandoned or Turn-Exit</td>
<td>%</td>
<td>So, -</td>
</tr>
<tr>
<td>Appreciation</td>
<td>ba</td>
<td>I can imagine.</td>
</tr>
<tr>
<td>Yes-No-Question</td>
<td>qy</td>
<td>Do you have to have any special training?</td>
</tr>
<tr>
<td>Non-verbal</td>
<td>x</td>
<td>[Laughter], [Throat_clearing]</td>
</tr>
<tr>
<td>Yes answers</td>
<td>ny</td>
<td>Yes.</td>
</tr>
<tr>
<td>Conventional-closing</td>
<td>fc</td>
<td>Well, it's been nice talking to you.</td>
</tr>
<tr>
<td>Uninterpretable</td>
<td>%</td>
<td>But, uh, yeah</td>
</tr>
<tr>
<td>Wh-Question</td>
<td>qw</td>
<td>Well, how old are you?</td>
</tr>
<tr>
<td>No answers</td>
<td>nn</td>
<td>No.</td>
</tr>
<tr>
<td>Response Acknowledgement</td>
<td>bk</td>
<td>Oh, okay.</td>
</tr>
<tr>
<td>Hedge</td>
<td>h</td>
<td>I don't know if I'm making any sense or not.</td>
</tr>
<tr>
<td>Declarative Yes-No-Question</td>
<td>qy\d</td>
<td>So you can afford to get a house?</td>
</tr>
</tbody>
</table>
Exploring how artificial intelligence technologies could be leveraged to combat fake news.
Stance Detection involves estimating the relative perspective (or stance) of two pieces of text relative to an topic, claim or issue. For FNC-1 we have chosen the task of estimating the stance of a body text from a news article relative to a headline. Specifically, the body text may agree, disagree, discuss or be unrelated to the headline.

**FORMAL DEFINITION**

**Input**

A headline and a body text from two news articles.

**Output**

Classify the stance of the body text relative to the claim made in the headline.

Four classification categories:

1. Agrees: The body text agrees with the headline.
2. Disagrees: The body text disagrees with the headline.
3. Discusses: The body text discuss the same topic as the headline, but does not take a position
4. Unrelated: The body text discusses a different topic than the headline

**EXAMPLE HEADLINE**

"Robert Plant ripped up $800M Led Zeppelin reunion contract"
I have every publicly available Reddit comment for research. ~1.7 billion comments @ 250 GB compressed. Any interest in this?

I am currently doing a massive analysis of Reddit's entire publicly available comment dataset. The dataset is ~1.7 billion JSON objects complete with the comment, score, author, subreddit, position in comment tree and other fields that are available through Reddit's API.

I'm currently doing NLP analysis and also putting the entire dataset into a large searchable database using Sphinxsearch (also testing ElasticSearch).

This dataset is over 1 terabyte uncompressed, so this would be best for larger research projects. If you're interested in a sample month of comments, that can be arranged as well. I am trying to find a place to host this large dataset -- I'm reaching out to Amazon since they have open data initiatives.

EDIT: I'm putting up a Digital Ocean box with 2 TB of bandwidth and will throw an entire months worth of comments up (~5 gigs compressed) It's now a torrent. This will give you guys an opportunity to examine the data. The file is structured with JSON blocks delimited by new lines (\n).
Announcements

• Assignment 2: Text Classification
  – Due on Friday at noon

• Project Proposals
  – Due Friday next week

• Lectures:
  – Today: discussion of projects
  – Monday: more discussion of projects and project proposals
  – Wednesday: no lecture. I will be available to meet students in my office (2 to 3:20) to answer questions about projects and proposals
Examples of Projects from Past Years
# Projects on Document Classification

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis of financial journalism for market prediction</td>
<td>Reuters articles</td>
</tr>
<tr>
<td>Evaluating the accuracy of three distinct single-label text classification models</td>
<td>Reuters articles, and possibly Wikipedia</td>
</tr>
<tr>
<td>Reddit miner</td>
<td>Reddit posts</td>
</tr>
<tr>
<td>Document reading level classification</td>
<td>Reddit posts</td>
</tr>
<tr>
<td>Wikipedia pages and what pages are about cats</td>
<td>Wikipedia pages</td>
</tr>
<tr>
<td>Classification of documents by author</td>
<td>Gutenberg Books</td>
</tr>
</tbody>
</table>
# Projects on Text Generation/Simulation

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>The botinator chatbot system</td>
<td>IMDB</td>
</tr>
<tr>
<td>Author simulator 2015</td>
<td>General text</td>
</tr>
<tr>
<td>Poetry generation</td>
<td>General text</td>
</tr>
<tr>
<td>Text simulation</td>
<td>General text</td>
</tr>
<tr>
<td>Grammar-learning and sentence-generating AI</td>
<td>General text</td>
</tr>
</tbody>
</table>
# Projects on Product/Movie Reviews

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotten or fresh? An exploration on the freshness of movies</td>
<td>Rotten Tomatoes movie reviews</td>
</tr>
<tr>
<td>Rotten tomatoes movie review classification with machine learning and NLP</td>
<td>Rotten Tomatoes movie reviews</td>
</tr>
<tr>
<td>Analysis of different algorithms in classifying reviews</td>
<td>Product reviews</td>
</tr>
<tr>
<td>Reach for the stars: prediction of product review star ratings</td>
<td>Product reviews</td>
</tr>
<tr>
<td>Foodiecity</td>
<td>Facebook, Yelp, Twitter</td>
</tr>
</tbody>
</table>
## Projects on Sentiment Analysis

<table>
<thead>
<tr>
<th>Project Title</th>
<th>Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidence and sentiment analysis algorithms</td>
<td>Rotten Tomatoes movie reviews</td>
</tr>
<tr>
<td>News and stock price relationship via sentiment analysis and linear regression</td>
<td>News articles</td>
</tr>
<tr>
<td>Technology stock market predictions based on twitter trends</td>
<td>Twitter</td>
</tr>
<tr>
<td>Tonal analysis of tweets</td>
<td>Twitter</td>
</tr>
<tr>
<td>Sentiment analysis</td>
<td>Twitter</td>
</tr>
<tr>
<td>Sentiment analysis of trends on microblogs</td>
<td>Twitter and/or Facebook</td>
</tr>
<tr>
<td>BladeRunner, sentiment analysis on Twitter tweets</td>
<td>Twitter</td>
</tr>
</tbody>
</table>
# Projects on Other Topics

<table>
<thead>
<tr>
<th>Project Title</th>
<th>Data Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter trend detection</td>
<td>Twitter</td>
</tr>
<tr>
<td>Exploring spam filtering by applying various classifiers with SMS spam data sets</td>
<td>SMS texts</td>
</tr>
<tr>
<td>AutoCorrect</td>
<td>Any text</td>
</tr>
<tr>
<td>AlgorithmicQuestGenerator</td>
<td>Gutenberg Books</td>
</tr>
</tbody>
</table>
Slides from Students about Projects in Previous CS 175 Classes
### Accuracy with other categories of tweets as negative examples for sarcasm.

<table>
<thead>
<tr>
<th>Category</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Happy:</td>
<td>0.8594313175810974</td>
</tr>
<tr>
<td>Sad:</td>
<td>0.7597116539847817</td>
</tr>
<tr>
<td>Fearful:</td>
<td>0.8157366519470093</td>
</tr>
<tr>
<td>Courageous:</td>
<td>0.912745545911375</td>
</tr>
<tr>
<td>Sincere:</td>
<td>0.759090909090909</td>
</tr>
<tr>
<td>Relaxed:</td>
<td>0.8740629685157422</td>
</tr>
<tr>
<td>Stressed:</td>
<td>0.8192771084337349</td>
</tr>
</tbody>
</table>

**Table Legend:**
- **Left:** Label of the tweets that are mixed in with sarcastic tweets
- **Right:** Accuracy of the classifier with this particular mix of tweets.

Not bad for sarcastic!

![Graph showing classifier accuracy for different categories](image-url)
Jimmy Fallon tweet Simulation

wine for cats. Because every girl
dreams of hearing her man say 'nice
!!', for already getting

Thank you guys like having @
StephenAtHome and climbing a guy
raised $ 20 min: Man never heard
even realize they love you to the
photobomb with @

# WhyImSingle tweets helped #
LateNight # FunShow http:\/t . I spent
the return of The Year . # LateNight #
VladdyPootPoot ', b ' l .' Ha !!! London
# funtimes # BiYM # FallonTonight ', b

getting old, but whee did it again ! #
BurgerSummit # nomayo "', b ' Arnold @
Schwarzenegger showed me some of his other
reviews were bad.

need a little help Obama \xe2 \x9d l ( ft . Any opportunity to have no that guy
for the show about this many people
wearing a werewolf

he was arrested for stopping by the
day ? Keep tweeting the man say it
was throwing shade last night at 10
minutes . co / Ux3WCDbJG2 ', b '
Thanks Jon every post - Aime with a
baby carrots
Planning and Organization of Projects
Project Tips: Goals

• Be clear in your goals
  – e.g., “will systematically evaluate the accuracy of logistic regression and neural network classifiers on the Reuters data set and 2 other data sets”
  – Ok to not to have all the details of how you will get there, but important to know what the goal is
  – Ok if goals are updated/changed as you learn more about the problem

• Team members should agree on the goals
  – To be effective the team needs to be clear about the goals
  – If there is any doubt about what the goal is, ask questions and discuss
Project Tips: Plan in Stages

Plan your project in stages so that the overall project is not dependent on the riskier elements working.

Example:

PHASE 1

Original Documents → Standard Bag of Words → Standard Logistic Regression → Cross-Validation Experiments
Project Tips: Plan in Stages

Plan your project in stages so that the overall project is not dependent on the riskier elements working.

Example:

PHASE 1
Original Documents → Standard Bag of Words → Standard Logistic Regression → Cross-Validation Experiments

PHASE 2
Original Documents → Bag of Phrases (ngrams)
Project Tips: Plan in Stages

Plan your project in stages so that the overall project is not overly dependent on the riskier elements

Example:

PHASE 1
Original Documents → Standard Bag of Words

PHASE 2
Standard Bag of Words → Bag of Phrases (ngrams) → Standard Logistic Regression

PHASE 3
Bag of Phrases (ngrams) → Deep Neural Network → Cross-Validation Experiments
Project Tips: Evaluation Methods

• Very important to have a clear idea of how you will evaluate your system

• For some tasks, such as document classification, there are well-defined metrics that are straightforward
  – E.g., cross-validated classification accuracy

• For other tasks, such as clustering, you will have to do some research to figure out what metrics are appropriate
  – For some projects, some user evaluation may be necessary

• Always include a baseline method in your experiments
  – E.g., for classification your baseline could be a Naïve Bayes classifier
Project Tips: Revision/Source Control for Code

• Each team should use a revision control system
  – e.g., Github system (freely available)
  – If you are not familiar with these systems, this is a good time to learn

• Revision control
  – Provides a systematic way for a team to develop code, scripts, documents, etc
  – Individuals can “check out” code, work on it, and then “commit”
  – Earlier versions of code can be recovered
    • Useful when you want to go back to an earlier version without a bug
Project Tips: Revision Control for Experiments

• You are likely to conduct many experiments over the course of the project, comparing versions of preprocessing, parameter settings, algorithms
  – You want to be able to keep track of your experiments and results in a systematic way

• Recommendations:
  – Organize and document your code/scripts for experiments
  – Use time-stamps, give your scripts interpretable names, use comments

• Consider using tools such as IPython Notebook, with github, for documentation and collaboration
  – just like a “lab notebook” in a science lab

• Reproducibility is important
  – Scripts will allow to you to regenerate results from earlier in the project
  – Also allows team members to share results efficiently
General Project Tips: Writing

• Write clearly
  – Try to put yourself in the mind of the person reading it
  – Make sure you don’t leave out important details and concepts
  – Use spell-checkers, grammar-checkers, etc
  – Target audience = a random student in the class

• Work collaboratively
  – Teams can use a shared document with version control, e.g., Google Docs
  – Have each member of the team edit, read, comment on the document
  – Work on your documents collaboratively: will lead to better documents

• Use figures and tables when you can
  – “A picture is worth a 1000 words”
Schedule for Next Week

• Monday
  – Discussion of project proposals in detail
  – Additional ideas for projects

• Wednesday
  – No lecture
  – I will meet students in my office (DBH 4216) to discuss project proposals
    • E.g., if you have a draft proposal I will review it quickly and give you feedback
    • E.g., if you need help with part of your proposal
    • E.g., if you are trying to figure out what data set to us.
    • And so on....