# CS 410 Project Topics

## Overview

**We will use Microsoft CMT to manage course project grading. Each student should please create an account there using their illinois email ids.**

**After deciding your topics, each student should please enter their details in this sign-up sheet**. Carefully enter the information used while registering in CMT into the first few columns *(this information will be used for grading, so please be careful!*). Then, enter your group name (could be anything) and project topic. Only the group leader needs to enter the project topic. However, every student needs to enter all other details.
**This needs to be completed before the proposal submission, i.e. before Oct, 25 to facilitate grading.**

**Multiple groups can choose the same topic.** Feel free to coordinate with other groups working on the same topic. For example, different groups can work on separate sub-tasks to increase the project-scope and overall contribution.

For the course project topics, we provide five broad categories of options for you:

1. You can choose to **reproduce the model and results in a published paper**. We provide some papers below. If you choose one of those papers, your project proposal is almost certain to get "approved"*.
2. You can choose to **improve over a current system by adding a function that is relevant to this course**. We provide some systems and candidate functions to add below. If you choose among those systems and functions, your project proposal is almost certain to get "approved"*.
3. It is possible that you **work on other papers or systems** that are not listed by us.
4. You can choose to **join a text classification competition, or an information retrieval competition**. If you choose this option, your project proposal is almost certain to get "approved"*.
5. You can **freely propose a topic relevant to this course**.

* For all the categories, the instructors will carefully review your project proposals and provide feedback. If we find your project topic/plan has some limitations, we will provide suggestions to improve it or suggest you to pick one of the sample topics. You're allowed to change topics after the proposal stage based on our feedback.

More detailed information about each option is given below.

# Option 1: Reproducing A Paper

You can choose to reproduce one of the following papers from one of the following subtopics:

- Subtopic: Latent aspect rating analysis
  - Hongning Wang, Yue Lu, and ChengXiang Zhai. 2011. Latent aspect rating analysis without aspect keyword supervision. In Proceedings of ACM KDD 2011, pp. 618-626. DOI=10.1145/2020408.2020505
- Subtopic: Pattern annotation
  - Qiaozhu Mei, Dong Xin, Hong Cheng, Jiawei Han, and ChengXiang Zhai. 2006. Generating semantic annotations for frequent patterns with context analysis. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2006). ACM, New York, NY, USA, 337-346. DOI=10.1145/1150402.1150441
- Subtopic: Contextual text mining
  - ChengXiang Zhai, Atulya Velivelli, and Bei Yu. 2004. A cross-collection mixture model for comparative text mining. In Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2004). ACM, New York, NY, USA, 743-748. DOI=10.1145/1014052.1014150
  - Qiaozhu Mei and ChengXiang Zhai. 2006. A mixture model for contextual text mining. In Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2006). ACM, New York, NY, USA, 649-655. DOI=10.1145/1150402.1150482
  - Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In Proceedings of the 15th international conference on World Wide Web (WWW 2006). ACM, New York, NY, USA, 533-542. DOI=10.1145/1135777.1135857
- Subtopic: Causal topic modeling
  - Hyun Duk Kim, Malu Castellanos, Meichun Hsu, ChengXiang Zhai, Thomas Rietz, and Daniel Diermeier. 2013. Mining causal topics in text data: Iterative topic modeling with time series feedback. In Proceedings of the 22nd ACM international conference on information & knowledge management (CIKM 2013). ACM, New York, NY, USA, 885-890. DOI=10.1145/2505515.2505612

All these papers are discussed in the lectures of Week 12. Once you have chosen one of these papers, please provide clear answers to the following questions in your proposal:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. Which paper have you chosen?
3. Which programming language do you plan to use?

4. Can you obtain the datasets used in the paper for evaluation?
5. If you answer “no” to Question 4, can you obtain a similar dataset (e.g. a more recent version of the same dataset, or another dataset that is similar in nature)?
6. If you answer “no” to Questions 4 & 5, how are you going to demonstrate that you have successfully reproduced the method introduced in the paper?

At the final stage of your project, you need to deliver the following:

- Your documented source code and main results.
- A demo that shows your code can actually run on the test dataset and generate the desired results. You don’t need to run the training process during the demo. If your code takes too long to run, try to optimize it, or write some intermediate results (e.g. inverted index, trained model parameters, etc.) to disk beforehand.
- Discuss how your results match or mismatch those reported in the original paper. Your results should cover all the main aspects and datasets discussed in the paper.
- If some of your results do not match the paper, discuss possible reasons and solutions.

# Option 2: Improving A System

You may choose to improve a system or service. We provide some candidates below. **Depending on your group size and the complexity of the techniques you develop, you may choose to work on one or several sub-topics provided per system.** Again, it should take you at least 20*N hours, where N is the total number of students in your team.

## 2.1 MeTA Toolkit

By now, you should all be familiar with the MeTA toolkit and its Python library Metapy. You can also refer to the publication. Choose this option if you wish to contribute to it as stated below.

- *Enhance MeTA and Metapy usability*

  Many of you have experienced difficulties while using Metapy in assignments this semester. You now have a chance to improve it so that future students and researchers can use this useful resource easily! Some ideas for improvement are given below:

  - Make Metapy compatible with the latest Python versions and different OS systems
  - Integrate it with existing popular toolkits e.g. NLTK, gensim
  - Enhance available tutorials for installing and using the tool on different platforms

- *Add text mining functions to the MeTA toolkit*

  The aim of this subtopic is to add some existing text mining algorithms to MeTA. Note that the papers mentioned below are all discussed in the lectures of Week 12 but cannot go to Option 1 because there are online GitHub repositories that implement them. If you want to borrow some code snippets from some repositories, make sure the licenses of those repositories allow you to do so, and you should follow the instructions in the licenses. If a repository does not have a license file, according to GitHub, the default copyright laws apply, meaning that the authors retain all rights to their source code and no one may reproduce, distribute, or create derivative works from their work. That means you CANNOT use codes from GitHub repositories without a proper license unless you obtain explicit written permission from the authors.

- Latent aspect rating analysis, given by the following paper
  - Hongning Wang, Yue Lu, and ChengXiang Zhai, Latent aspect rating analysis on review text data: a rating regression approach. In Proceedings of ACM KDD 2010, pp. 783-792, 2010. DOI=10.1145/1835804.1835903
- Topic modeling with network regularization, given by the following paper
  - Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. 2008. Topic modeling with network regularization. In Proceedings of the 17th international conference on World Wide Web (WWW 2008). ACM, New York, NY, USA, 101-110. DOI=10.1145/1367497.1367512

## 2.2 ExpertSearch System

The ExpertSearch system (http://timan102.cs.illinois.edu/expertsearch//) was developed by some previous CS410 students as part of their course project! The system aims to find faculty specializing in the given research areas. The underlying data and ranker currently comes from the MP2 submissions of the previous course offering. You can read more about it here (Sections 3.6 and 4: Project are especially relevant). The code is available here. Below are some ideas to improve and expand this system. You may choose to integrate your code with the existing system, or borrow some ideas from it, or build your own systems/algorithms from scratch.

- *Automatically crawling faculty webpages*

  Recall that you developed scrapers for faculty web-pages in MP2.1, which, in general, can be a time-consuming task. So, the question is can we automate this process? Some challenges include:

  - **Identifying faculty directory pages:** First, we need to identify the pages from where faculty web-pages can be mined. In MP2.1, we used faculty directory pages as the starting point to find faculty webpages. So, given a university website, can we automatically identify the directory pages? This can be posed as a classification task, i.e. classify a URL into a directory page vs. non-directory page. We have a huge resource of directory page URLs available in the sign-up sheet. These can be the “positive” examples. You can get a list of some random URLs online or crawl some other pages to get URLs(e.g. other URLs on the university websites, product websites, news sites,etc.). These would be the “negative” examples.

  - **Identifying faculty webpage URLs:** Next, we need to extract the faculty webpages from the directory pages. This can again be posed as a classification task. Given a URL, can we identify whether it is a faculty webpage or not? We have a huge resource of faculty webpage URLs (available under MP2.3 on Coursera). These would be the “positive” examples. You can get a list of some random URLs online or crawl some other pages to get URLs (e.g. other URLs on the university websites, product websites, news sites, etc.) to get the “negative” labels.

- *Extracting relevant information from faculty bios:*

  The problem here is to convert the unstructured text in faculty webpages into more **structured text**. Such structured information would enhance the utility of the system. For example, in the ExpertSearch system, emails and faculty names extracted from bios are shown in the search results. Users can click on the “mail” button to directly mail the faculty. Extraction is done using regex-based techniques and Named Entity Recognition (NER) that don’t always work well. Can you improve those existing techniques?
  You can also develop techniques for extracting other information, e.g. faculty research interests. For example, you may perform *topic mining* on the bios available under MP2.3 on Coursera. The top-keywords per topic could be the common research areas. You might also perform *keyword extraction* from faculty bios, research papers, etc.

## 2.3 EducationalWeb System

The EducationalWeb system (http://timan102.cs.illinois.edu/explanation//slide/cs-410/0) is a tool to help students learn from course slides. It has two main functionalities currently: 1) Retrieve and recommend relevant slides for each slide. You can read more about this in the following papers Web of Slides, WOS Demo.; 2) Find an explanation of a term/phrase on the slide by highlighting it and then clicking on the "cap/scholar" button on the top-right of a slide. It will try to retrieve a relevant section from the Professor's textbook that contains an explanation of the selected phrase. You can read more about the underlying algorithm here.

The code for the system is available here. Below are some ideas to improve and expand this system. You may choose to integrate your code with the existing system, or borrow some ideas from it, or build your own systems/algorithms.

- *Improving the usability and reach of the existing system*

  Some of you might have used the system and identified potential areas of improvement. The aim of this subtopic is to refine the current version of EducationWeb. Some specific ideas include (many are borrowed from this Piazza post):

  1. **Scale up the current system**. Add more slides and courses from multiple sources e.g. Coursera, UIUC courses, etc. and run the existing algorithms on them. Again, it might be useful to think about automatic crawling similar to the subtopic in 2.2 above. It would be very interesting to see the interaction between slides/textbooks at a large scale!!
  2. **Improve the performance of the system**. Currently, loading each slide takes time.
  3. **Allow downloading slides in bulk.** Currently, we can only download one slide at a time.
  4. **Add more context to the explanations** (e.g. link to the specific page in the textbook)
  5. **Allow adding additional courses/lectures directly from the web interface**. This would also involve dynamically identifying the recommended/relevant slides for a new slide. Currently, a static file is used which contains pre-computed recommendations for each slide.
  6. **Integrate the tool with Piazza/Coursera**, i.e. maybe link Piazza/Coursera to the tool or vice-versa. Alternatively, add discussion forum and video capabilities to the tool so that it serves as a one-stop-shop for all users' educational needs.
  7. **Link to latest related research articles**: In this way, the lecture content can be automatically updated
  8. You could also work on **improving** the current recommendation, search and explanation mining **algorithms** (described in the papers at the beginning of this section 2.3)

- *Automatically creating teaching material for in-demand skills*

  This subtopic is an extended version of the existing EducationWeb system. There is an increasing demand for skilled workers in the industry. Quality education is not easily accessible to everyone due to barriers such as high cost, geographical and language barriers, etc. Also, instructors cannot be available 24*7 to provide personalized support to all learners. In this subtopic, the overarching aim is to tackle some of these issues. In particular, the following tasks might be good starting points.

- **Identifying in-demand skills:** You can crawl and analyze relevant sections of job boards, news articles, scientific articles, social media, etc. to automatically identify the *emerging keywords*/topics. For this, you may refer to some papers on contextual text mining (mentioned in Option 1 of this document).

- **Creating lectures and tutorials for those skills:** For this, you may consider lecture slides (e.g. from Coursera courses) as the basic units of knowledge. Then, the task could be to find the most relevant slides or clusters of slides (could be across multiple courses/lectures) for a given skill (topic). You may borrow some ideas from the EducationWeb system for this.
  You may also use the slides in existing lectures on some topics as the "relevant slides" for those topics. In this way, you can automatically generate training data for supervised learning.

  You could also combine knowledge from multiple sources (e.g. textbook sections, slides, videos, blogs, codebases) for creating more comprehensive tutorials.

  A more challenging task would be to automatically *generate* the lectures/tutorials using techniques from natural language generation and abstractive text summarization. Another interesting idea is to automatically *generate agents*, e.g. using Virtual Agent Interaction Framework (VAIF). This goes beyond the material covered in class but could lead to some highly innovative and state-of-the-art projects!

If you choose this option, please answer the following questions in your proposal:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. What system have you chosen? Which subtopic(s) under the system?
3. Briefly describe the datasets, algorithms or techniques you plan to use
4. If you are adding a function, how will you demonstrate that it works as expected? If you are improving a function, how will you show your implementation actually works better?
5. How will your code communicate with or utilize the system? It is also fine to build your own systems, just please state your plan clearly
6. Which programming language do you plan to use?
7. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

At the final stage of your project, you need to deliver the following:

- Your documented source code.
- A demo that shows your implementation actually works. If you are improving a function, compare your results to the previously available function. If your implementation works better, show it off. If not, discuss why.

# Option 3: Working on Other Papers or Systems

It is also possible that you work on other papers or systems that are not listed by us. Because we do not list them, we will need more information from your proposal to decide whether you have a good topic. You may find more guidelines below.

## 3.1 Reproducing an unlisted paper

If you choose to reproduce a paper not listed under Option 1, please make sure that your chosen paper satisfies the following criteria:

1. The paper should solve one of the research challenges introduced in lectures.
2. The paper should be published at a trustable venue (i.e. conference or journal). Some examples include ACM SIGIR, KDD, EMNLP, ACL, Learning @ Scale, EDM, etc. If you can find Cheng's paper(s) at some venue, then that venue is likely to be trustable.
3. There should be NO publicly available implementation for this paper. For example, if the main method in a paper is already released on GitHub or built into a library, you cannot choose that paper.
4. The main method in the paper should be advanced enough so that the work of reproducing it likely takes at least 20*N hours, where N is the total number of students in your team. You may justify this by listing the main tasks to be completed and the estimated time cost for each task.

In your proposal, please explain how your chosen paper satisfies the above criteria, and also answer the questions listed under Option 1. Other requirements are the same as Option 1.

## 3.2 Improving over a paper

You can choose to improve over a paper that is relevant to one of the tasks introduced in the lectures. If you choose this option, please answer the following questions in your proposal:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. What paper have you chosen?
3. What is your idea for improving the paper/system? Why do you think your idea will hopefully work better?
4. How are you going to evaluate your idea? What are the datasets and baseline methods?
5. Which programming language do you plan to use?
6. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

At the final stage of your project, you need to deliver the following:

- Your documented source code and main results.

- A demo that shows your code can actually run on the test dataset and generate the desired results. You don’t need to run the training process during the demo. If your code takes too long to run, try to optimize it, or write some intermediate results (e.g. inverted index, trained model parameters, etc.) to disk beforehand.
- If your idea works, discuss what advantages it has over the original paper, and what possible limitations are still there in your method. If your idea does not work, don’t worry - just discuss possible causes and potential solutions, and you will get your credits as long as your study is solid and your discussion is thorough.

## 3.3 Adding an unlisted function to a listed system

You can choose to improve one of the systems listed under Option 2 by adding a function that is not listed there but relevant to the course content. If you choose this option, please answer the following questions in your proposal:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. What system have you chosen? What function are you adding? How will the new function benefit the users?
3. How will you demonstrate that the new function works as expected?
4. How will your code communicate with or utilize the system?
5. Which programming language do you plan to use?
6. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

At the final stage of your project, you need to deliver the following:

- Your documented source code. Explain how your code communicates with or utilize the system.
- A demo that shows your implementation actually works.

## 3.4 Improving over an unlisted system

You can choose to improve a system or service that is relevant to the course content but not listed under Option 2. You may either add a new function to the system or improve the performance of an existing function. If you choose this option, please answer the following questions in your proposal:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. What system have you chosen? Are you adding a function or improving a function? What function?
3. If you are adding a function, why is the new function important or interesting? How will it benefit the users? If you are improving a function, what are the main limitations of the

current function? How are you going to improve it? How will your improvements benefit the users?
4. If you are adding a function, how will you demonstrate that it works as expected? If you are improving a function, how will you show your implementation actually works better?
5. How will your code communicate with or utilize the system?
6. Which programming language do you plan to use?
7. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

At the final stage of your project, you need to deliver the following:

- Your documented source code. Explain how your code communicates with or utilize the system.
- A demo that shows your implementation actually works. If you are improving a function, compare your results to the previously available function. If your implementation works better, show it off. If not, discuss why.

# Option 4: Competitions

**This option may fit you well if you would like to gain some experience in state-of-the-art text classification or information retrieval practices**. You will need to research by yourselves some cutting-edge models that are more recent than those introduced in the lectures. Of course your TAs will be there for you when you need help. Performance is the most important factor. Once you achieve the state-of-the-art performance, we give you a bonus (10% extra credit) that can be used to cover any loss of points in the project caused by small mistakes. We hope you can have fun in learning and trying recent methods in this option.

We are thinking of hosting an **information retrieval (IR) competition** and a **text classification competition**. You will have access to the text and labels of a training dataset and the text of the test dataset, but you cannot see the labels of the test set. You will then develop a text classifier or a document ranker and submit your test set predictions. We will automatically evaluate your results on the test dataset and release the test set performance of all participating teams on a leaderboard (the setup would be similar to MP2.4).

**The text classification competition is available here. It is also available on LiveDataLab along with the baseline scores.**

**The IR competition is available here. It is also available on LiveDataLab along with the baseline scores.**

You're allowed to use pre-built machine learning packages. However, if you find someone's solutions (source code) to similar competitions online, you may not use them directly. You're free to borrow some of their ideas with proper citations and credit attribution. You can also use publicly available external datasets but please make sure they don't overlap with the test sets.

In each competition, you will compete with a competitive baseline and your classmates.

Your grade will largely depend on your test set performance. Recall that in your project grading, there is **45% on "source code submission"**. (You can find the project grade composition in

Week 1.) Assuming that there is no issue with your submitted source code, that 45% will be graded with the following criteria based on your test set performance on the leaderboard:

- **If you outperform the baseline, you get all 45%, plus 10% extra credit** to make it up for you if you lose points in other parts of the project. The extra credit will not make you earn more than 100% for your project, and cannot be applied to other parts of this course.
- If you do not outperform the baseline but make a valid submission, you get 15% + 30% * (1 - (r-1)/N), where r is your rank on the leaderboard and N is the total number of teams that have chosen this option.

If there is an issue with your source code (e.g. using others' code without properly handled copyright, obvious bugs, etc.), it is possible that you get a lower score than those listed above.

In your project proposal, please answer the following questions:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. Which competition do you plan to join?
3. If you choose the IR competition, are you prepared to learn state-of-the-art IR methods like query expansion, feedback, rank fusion, learning to rank, etc.? Name some more concrete methods or tools that you may have heard of. If you choose the classification competition, are you prepared to learn state-of-the-art neural network classifiers? Name some neural classifiers and deep learning frameworks that you may have heard of. Describe any relevant prior experience with such methods
4. Which programming language do you plan to use?

At the final stage of your project, you need to deliver the following:

- Your documented source code and test set predictions.
- Explain your model, and how you perform the training. Describe your experiments with other methods that you may have tried and any hyperparameter tuning.
- A demo that shows your code can actually run on the test set and generate your submitted predictions. You don't need to run the training process during the demo. If your code takes too long to run, try to optimize it, or write some intermediate results (e.g. inverted index, trained model parameters, etc.) to disk beforehand.

# Option 5: Free Topics

You may freely propose a topic that is not listed in this document but relevant to this course. In your proposal, please answer the following questions:

1. What are the names and NetIDs of all your team members? Who is the captain? The captain will have more administrative duties than team members.
2. What is your free topic? Please give a detailed description. What is the task? Why is it important or interesting? What is your planned approach? What tools, systems or

datasets are involved? What is the expected outcome? How are you going to evaluate your work?

3. Which programming language do you plan to use?
4. Please justify that the workload of your topic is at least 20*N hours, N being the total number of students in your team. You may list the main tasks to be completed, and the estimated time cost for each task.

At the final stage of your project, you need to deliver the following:

- Your documented source code and main results.
- Self-evaluation. Have you completed what you have planned? Have you got the expected outcome? If not, discuss why.
- A demo that shows your code can actually run and generate the desired results. If there is a training process involved, you don’t need to show that process during the demo. If your code takes too long to run, try to optimize it, or write some intermediate results (e.g. inverted index, trained model parameters, etc.) to disk beforehand.