CMSC 673 Graduate Assessment, Fall 2022

CMSC 673 — Introduction to Natural Language Processing

Item	Summary	Points
Checkpoint 1	Monday October 3rd	10
Checkpoint 2	Friday November 4th	25
Checkpoint 3	Friday November 18th	65
Final turn-in	Friday December 9th	100

All due dates are 11:59 PM Baltimore time.

For the core graduate assessment (worth 40% of the CMSC 673 grade), choose do to **ONE** of the following assessments, (GA1)–(GA5). Four of them will require you to implement NLP algorithms. One of them is to write a 4 page (conference-style) literature review paper on a particular topic. While each person is free to choose their own assessment, I encourage each person to strongly consider how each may be able to further their career/degree goals and choose the assessment that maximizes progress toward their goal. For example, someone writing a disseration (doctoral or masters) may wish to do the literature review paper, since it could help form the basis of the related work chapter of a dissertation. On the other hand, someone may wish to really dive deep into an aspect of NLP implementation.

The possible assessments are divided into an "Implementation Track," and a "Paper Track":

Implementation Track

(GA1) Implement a linear chain conditional random field and evaluate it on a sequence tagging task, such as named entity recognition, part-of-speech tagging, syntactic chunking, or a task of your choice.

(GA2) Implement the inside-outside and CKY algorithms for probabilistic context free grammars, and evaluate it on semi-supervised syntactic parsing.

(GA3) Using the UD corpora, implement and evaluate a neural arc-standard dependency.

(GA4) Implement (smoothed) Latent Dirichlet Allocation and evaluate it quantitatively via language modeling perplexity. Qualitatively examine and analyze the resulting topics.

Paper Track

(GA5) Write a literature review paper in which you select one of a set of topics, and (i) identify, (ii) analyze, and (iii) synthesize modern approaches for the topic you choose.

These are all advanced topics that we *may* very lightly cover, or mention, in the course. However, we will not spend significant time on any of them. In all cases, the only reference materials you may peruse, read, skim, look at, use, or otherwise consult are

(Y1) published textbooks (online versions or drafts are acceptable);

(Y2) peer-reviewed papers;

(Y3) arXiv preprints;

(Y4) slides from tutorials or presentations for papers given at conferences;

(Y5) course materials from this class (such as slides, lecture discussion, and Discord notes and discussion);

(Y6) general-purpose computing libraries (numpy, scipy, blas, etc.);

(Y7) the evaluation modules/functions from scikit-learn.

You may receive sanctioned UMBC writing assistance, such as through the GSA Writing Advisor.[1]

You are **not** allowed to peruse, read, skim, look at, use, or otherwise consult any of the following:

(N1) blog posts;

(N2) any sort of article, writing, manuscript, or slides not covered by the "Y" list (unless specifically approved in advance, in writing);

(N3) code or pseudo-code available online;

(N4) machine learning frameworks like scikit-learn (except for the evaluation modules), PyTorch, or Tensorflow (except for options (GA3) and (GA5)) [this list is not inclusive];

(N5) other online courses and their materials.

The use of any of the "N" items will be considered a serious academic integrity violation and will result, *at a minimum*, in an automatic 0 on this entire assessment (i.e., a 0 will be recorded for 40% of your entire course grade).

The rest of this document is divided into two main sections: one describing the "Implementation Track" tasks, milestones, and deliverables; and the other describing the "Paper Track" tasks, milestones, and deliverables. All options have four components: three checkpoints and the final turn-in. All checkpoints share the same due dates. While what is required for the milestones depends on whether you choose the implementation track or paper track, the first checkpoint/milestone is the same for all options.

The following table summarizes what is due when:

Item	**If you select from the Implementation Track:**	**If you select from the Paper Track**	**Due**
Checkpoint 1	Selection of one of (GA1)–(GA4), and initial list of resources	Selection of topic for (GA5), and initial list of papers	Monday October 3rd
Checkpoint 2	Initial report discussing progress, hurdles, and other challenges; and git repo containing at least three non-trivial/starter commits	Complete first draft and paper source	Friday November 4th
Checkpoint 3	Feedback/suggestions on another student's initial report	Review	Friday November 18th
Final turn-in	Completed code, full git repository, completed writeup, and document summarizing how the feedback was used	Completed final version, paper source, and summary-of-changes document	Friday December 9th

[1] `https://gsa.umbc.edu/writing-advisor/`

1 Implementation Track: Assessment Options (GA1)–(GA4)

For any of the options (GA1)–(GA4), you must

(i) implement the specific algorithm or system,

(ii) train it on the specified data,

(iii) evaluate it on the specified data,

(iv) create a written lab report/writeup that includes

   (a) a prose-based discussion of the algorithm;
   (b) a prose-based discussion of your implementation, including any hurdles you encountered (and how you addressed them);
   (c) a discussion of what tests you ran to ensure correctness;
   (d) results on both dev/validation and test splits; and
   (e) a summary of the approach's ability to do the task(s) you evaluated.

Be sure to cite appropriately and follow all academic honesty standards. Accounting for items like tables and graphs, an appropriate target length of this writeup is anywhere from 1 to 3 pages (though that is not a hard limit).

Within the Implementation Track, you will be primarily evaluated on the completeness and correctness of your implementation. However, the thoroughness and clarity of the writeup will be a sizeable (but non-majority) portion of your grade.

1.1 Milestones for (GA1)–(GA4)

Any of the Implementation Track options have three milestones prior to the final submission. All milestones must be met for full credit.

Milestone 1: Selection of Option This is due Monday October 3rd by 11:59 PM. Select which of the options (GA1)–(GA4) you will do, and identify between 3 and 7 approved resources you think may be valuable to use in your work. Record your option selection, and upload a PDF list of the approved resources to the submission site, `https://www.csee.umbc.edu/courses/undergraduate/473/f22/submit`, selecting "GA: Milestone1." This list should not be your complete or final list of all resources you might use: it is meant as a starting point. You may also remove papers/resources from this list as you make progress.

Milestone 2: Initial draft of writeup, and git repo of your code with ≥ 3 non-trivial commits This is due Friday November 4th by 11:59 PM. You must submit two items: an initial version of your writeup, and a git repository of your implementation containing at least three, non-trivial commits (e.g., commits affecting more than just whitespace or comments). Your writeup does not need to include results, but it does need to include a prose-based discussion of the algorithm; a prose-based discussion of your implementation (or expected implementation), including any hurdles you are encountering; and a discussion of what tests you are running or will run to ensure correctness. Submit via the submission site, `https://www.csee.umbc.edu/courses/undergraduate/473/f22/submit`, selecting "GA: Milestone 2." You must turn in:

- an **ANONYMIZED** PDF of your writeup,
- a git repo for your code.

Your anonymized PDF will be provided to other students to review/provide feedback on (see Milestone 3); your code will not be shared. You may *optionally* provide a written description of what, if any, writing assistance you received (e.g., the GSA Writing Advisor).

Milestone 3: Feedback on Discussion This is due Friday November 18th by 11:59 PM. In this process, you will receive up to two other students' initial writeups; you must provide feedback on the breadth, depth, and clarity of exposition. You may also provide suggestions on hurdles that are described in the writeup(s). Reviewing forms and guides will be provided. To receive full credit for the reviews, you must provide constructive and civil reviews (a guide will be provided).

This feedback will be "double-blind:" as a reviewer, you will not know whose writeups you are reviewing, and as an author, you will not know who your reviewers are. This is why it is important for the Milestone 1 drafts to be anonymized. All paper-reviewer identities will be known to course staff.

Final Writeup and Full Code This is due Friday December 9th by 11:59 PM. This must be a **complete**, well-written writeup. These should be submitted to the submission site, `https://www.csee.umbc.edu/courses/undergraduate/473/f22/submit`, selecting "GA: Final Turn-in." You must turn in:

- a **NON-ANONYMIZED** PDF of the writeup,
- your code repo,
- a PDF document discussing the changes made, both as a result of the reviews/feedback and along with any unprompted changes.

As with the initial submission, you may also provide a written description of what, if any, writing assistance you received (e.g., the GSA Writing Advisor).

1.2 Specifics of Each of Implementation Track Option

(GA1): Implement a CRF For this option, implement a supervised linear chain conditional random field, as described in Lafferty, J. D., McCallum, A., and Pereira, F. C. N. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data" (ICML 2001). You must implement this CRF from scratch, and train and evaluate it on an appropriate sequence tagging task, such as POS tagging, named entity recognition (NER), or syntactic chunking. You can substitute a different gradient optimizer from the one they originally describe. The precise task is your choice: if doing POS tagging, you may use the UD data, though the instructor may need to provide separate data if you're interested in another task (like NER).

(GA2) Implement the inside-outside and CKY algorithms for probabilistic context free grammars. For probabilistic CKY, see chapter 14.2 of 3SLP. For inside-outside, while you may reference any resource in compliance with the "Y" resources in the beginnning, you may also reference Michael Collins's notes on the inside-outside algorithm: `http://www.cs.columbia.edu/~mcollins/io.pdf`. Consulting these notes will not be an academic integrity violation. Use the inside-outside algorithm to train in a semi-supervised manner (using the Penn Treebank sections 02-21 for supervised data, and the UD English EWT training sentences for unsupervised data), and evaluate on the Penn Treebank (dev evaluation on section 22, test evaluation on section 23). Obtain parses using your CKY implementation, and evaluate using the publicly available `evalb` script (`https://nlp.cs.nyu.edu/evalb/EVALB.tgz`).

(GA3) Implement and evaluate a neural arc-standard dependency parser. The algorithm is described in 3SLP or the Eisenstein book, and you may follow the broad outline given by Chen and Manning (2014, EMNLP), but note that you do not have to re-implement this work exactly. I *strongly* recommend that you run through the algorithm on paper, and really understand it, before you begin coding. Train and evaluate your system twice: the first time on English-EWT, the other is another language from UD of your choice. For this option, you may use the toolkits Pytorch, Tensorflow, or Keras. You may

(GA4) Implement the smoothed LDA model from Blei et al. (2003). You may follow their approach of doing variational inference, though if you're aware of, or want to learn collapsed Gibbs sampling, you are welcome to implement that instead. Evaluate in two ways: first, quantiatively via heldout perplexity. Second, qualitatively analyze the topics: do they "make sense?" Do they appear coherent? Train and evaluate your implementation on two different UD datasets (one English, one not).

2 Paper Track: Assessment Option (GA5)

Select one of the following topics (see Section 2.2), and (i) identify, (ii) analyze, and (iii) synthesize modern approaches for the topic you choose.

Identify For this assignment you will need to find an appropriate number of papers to discuss in detail. Though the final number that you select is highly dependent on, among other things, which topics you choose, the length of the papers, and their venues, a reasonable number of papers is between five and ten. This range does not constitute required minimums or maximums.

You may read many more papers than you discuss in detail. Do not view this as "wasted" effort—these should help inform the overall narrative and context for your discussion.

Analyze Ask and answer fundamental research questions: what were the goals of each of the papers? What scientific and engineering questions did each of the tackle? How well did the evaluations support the main claims? What was not done that could have been done?

Synthesize How do the efforts relate to one another? Do they follow one after another, making (incremental) progress on a task (metric)? Does one question some basic assumptions of another, and if so, how do the other papers fit in? What are the limitations of these approaches, and what still remains to be done? You can also **link these papers and ideas to related fields**.

Within the Paper Track, you will be primarly evaluated primarily on the completeness, thoroughness, and clarity of your paper. Grammatical, logical, organizational, or factual errors will be negatively impact the score.Weak or lacking analysis and synthesis will also be large negative influences on the score.

Requirements Papers should be four pages, not including references, in the ACL format. Paper **must** use the ACL 2018 style guide; both LaTeX and Microsoft Word (docx) versions are on GL:
`/afs/umbc.edu/users/f/e/ferraro/pub/sty/acl18/acl18-latex.zip`
`/afs/umbc.edu/users/f/e/ferraro/pub/sty/acl18/acl18-word.zip`

Be sure to cite appropriately and follow all academic honesty standards. You may include figures (your own, reproductions, or copies of existing figures); be sure to provide appropriate credit for the figures. However, make the figures count: do not include them simply to pad the paper. Do not consider just "recent" papers; try to find papers from the past 25 years.

2.1 Milestones for (GA5)

This paper has three milestones prior to the final submission. All milestones must be met for full credit.

Milestone 1: Selection of Option, Topic, and Initial list of papers This is due Monday October 3rd by 11:59 PM. Decide on a topic and compile a list of at least five relevant papers. You do not need to have read the papers but you *should* have read the abstracts. Your topic choice and list of papers (in a PDF) should be submitted to the submission site. This list should not be your complete or final list of papers you'll read or consult: it is meant as a starting point. You may also remove papers from this list when you actually write your paper.

Milestone 2: Initial version of the paper This is due Friday November 4th by 11:59 PM. Despite it being "initial," this must be a **complete**, well-written paper. Although this submission will not solely comprise your grade, it should be a paper that you would find acceptable for determining your grade for the course's paper component. To receive full credit, your paper must be a legitimate and full response to the prompt. Submit via the submission site, `https://www.csee.umbc.edu/courses/undergraduate/473/f22/submit`, selecting "GA: Milestone 2." You must turn in:

- an **ANONYMIZED** PDF of the paper,
- the paper's source (such that we could regenerate the PDF).

Your anonymized PDF will be provided to other students to review (see Milestone 3); your source will not be shared. You may also provide a written description of what, if any, writing assistance you received (e.g., the GSA Writing Advisor).

Milestone 3: Paper Peer Review This is due Friday November 18th by 11:59 PM. In this process, you will receive up to two other students' papers; you must provide feedback on the breadth, depth, and clarity of exposition. Reviewing forms and guides will be provided. To receive full credit for the reviews, you must provide constructive and civil reviews (a guide will be provided).

This review will be "double-blind:" as a reviewer, you will not know whose papers you are reviewing, and as an author, you will not know who your reviewers are. This is why it is important for the Milestone 1 papers to be anonymized. All paper-reviewer identities will be known to course staff.

Final version of the paper This is due Friday December 9th by 11:59 PM. This must be a **complete**, well-written paper. These should be submitted to the submission site, `https://www.csee.umbc.edu/courses/undergraduate/473/f22/submit`, selecting "GA: Final Turn-in." You must turn in:

- a **NON-ANONYMIZED** PDF of the paper,
- a PDF document discussing the changes made, both as a result of the reviews/feedback and along with any unprompted changes, and
- the paper's source (such that we could regenerate the PDF).

As with the initial submission, you may also provide a written description of what, if any, writing assistance you received (e.g., the GSA Writing Advisor).

2.2 Topics

Please select a topic from the three listed below. With consultation of the instructors, you may propose your own, separate topic.

2.2.1 Hierarchical/Advanced Language Modeling

For this topic, you will examine advanced and/or hierarchical approaches to language modeling. Examples include (modified) Kneser-Ney smoothing [Chen and Goodman (1999)], syntactic and/or semantic language models [Chelba and Jelinek (1998)], topic models [Blei et al. (2003)]; Teh et al. (2006)], hierarchical Bayesian language models [MacKay and Peto (1995)]; Teh (2006)] , and neural language models.

2.2.2 Grounded Language Processing

For this topic, you will examine how non-language signals (e.g., image or audio features) can help NLP tasks, how NLP tasks/models can improve understanding/analysis of those non-language signals, or both. For example, the task of (sequential) image captioning or video summarization involves producing natural language output that describes what is happening in those input images or videos. Visual question answering (and its variants) requires systems to answer questions about an image with a sunny sky in the background, answer the question "does it look like it's going to rain?" Meanwhile, tasks involving conversational and dialogue agents (e.g., Google Now, or Alexa) may need to take in spoken language input and perform some action (including generating "spoken" language output) based on it.

2.2.3 Structured Prediction for NLP

For this topic, you will examine *structured prediction* for a single task, or a significant, relevant aspect of that task. Roughly, structured prediction is any task that given an input, produces some object or label with an internal structure. This is in contrast to prediction tasks that simply predict a single, "flat" label, without any decomposable or introspective structure. We've already briefly talked about two instances of structured prediction—machine translation/alignment and part of speech tagging—and we'll see more as we get further along in the semester. Canonical examples of structured prediction problems include, but are not limited to: (a) syntactic parsing (constituency or dependency); (b) machine translation; (c) semantic parsing (including FrameNet, PropBank, AMR, and VerbNet parsing); (d) structured information extraction (such as template-based slot filling as in the ACE Relation Extraction task); (e) ontology induction; and (f) entity coreference or cross-document entity linking. Generally, "bags-of-items" models do not arise in structured prediction tasks. Tasks like question answering, recognizing/determining textual entailment, and sentiment analysis may or may not involve structured prediction.[2] Your paper is not restricted to the preceding items; they are offered as suggestions.

2.2.4 Computational Methods for Linguistic Subfields

For this topic you will examine how computational/statistical models are developed to better explain (or mimic) linguistic phenomena/subfields. For example, you could explore computational approaches to phonology, morphology, syntax, semantics, or pragmatics—or any combination, e.g., morphosyntax, syntactic-semantic, phonology/morphology, typology, etc. (Try searching the ACL Anthology for those terms.)

2.2.5 Natural Language Generation

For this topic you will examine methodological and/or evaluation approaches for generating natural language. Classic examples of natural language generation include machine translation and abstractive summarization. There's an entire SIG on generation (SIGGEN) and conference (INLG) devoted to it.

2.2.6 Ethical Issues and Bias in NLP

For this topic you will explore ethical concerns (and approaches for dealing with them) in NLP, and/or issues of implicit/explicit bias in NLP models. For example, this briefly appeared in some of the associations identified by Church and Hanks (1989). More recently you may have heard about how for a while Google Translate would always translate the Turkish sentence "o bir doktor" using the masculine pronoun "he" even though that Turkish sentence is gender neutral (this particular case has now been addressed). There are workshops on Ethics and gender bias in NLP. There is also a more general growing community called FAT/ML (Fairness, Accountability, and Transparency in Machine Learning). You can look through the associated proceedings (but make sure there's an NLP component).

2.2.7 NLP for `<Your Area>`

For this topic you would survey how NLP can be used in an area of study of interest to you. For instance, there are special interest groups (called SIGs) for NLP for the humanities, Semitic languages, and biomedical applications—among many others. Look at the "SIGs" row in the main table at a `https://aclweb.org/anthology/`.

[2] Although sentiment analysis often asks for a single label at the end, the internal classification may operate over structures. That is, the final flat label may be the result of some internal, latent structured prediction.

Workshops (`https://aclweb.org/anthology/venues/ws/`) also often offer targeted application and interest areas, for example in the legal domain, patents, economics, and others!

2.2.8 Your Choice!

Unsatisfied with any of the above options? Then feel free to pick your own topic. The requirement is that you *must* clear it with the instructor first and it must have a significant relevance to material covered in this course.

Where to Start

You may analyze any papers read in class or as part of the assignments. **You are welcome and encouraged to come talk with course staff, either during office hours, Discord, or by appointment to discuss topics, advice on finding relevant papers, and the direction of your paper.**

Google Scholar is an easy way to find linked and cited papers. Another great resource is the ACL Anthology (`http://aclanthology.info/`), which archives papers by conferences (e.g., ACL, EACL, NAACL, NAACL), journals (CL, TACL), and workshops by year.[3] It also offers multiple custom searches: for example, searching "distributed representations" returns papers for crosslingual word representations (C12-1089), representations for relational patterns (P16-1215), and representations for semantic role labeling (D15-1295). Note that in NLP, conferences, and even workshops, are preferred to journals; conference reviewing can be just as, if not more intense and selective, as journal reviews. NLP conferences and workshops are almost always peer-reviewed and archival (meaning they are "finished" publications).

The AAAI digital library also offers an extensive listing of AI-based conferences and proceedings. Of particular relevance are the flagship AAAI, ICML (International Conference on Machine Learning), and KDD (Knowledge Discovery and Data Mining) proceedings. Papers from NeurIPS (Neural Information Processing Systems) often tend to the more theoretical, but with a decided focus on neural networks.

The following is a *very* small listing of potential starting papers:

1. Hinton (1986): "Learning Distributed Representations of Concepts"
2. Brown et al. (1992): "Class-based n-gram models of natural language"
3. Rosenfeld (1994): "Adaptive Statistical Language Modeling: A Maximum Entropy Approach", Chapters 5-8
4. Kneser and Ney (1995): "Improved backing-off for m-gram language modeling"
5. Bengio et al. (2003): "A Neural Probabilistic Language Model"
6. Ando and Zhang (2004): "A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data"
7. Rosenfeld (2004): "Two Decades of Statistical Language Modeling: Where Do We Go From Here?"
8. Mnih and Hinton (2008): "A Scalable Hierachical Distributed Language Model"
9. Graves (2013): "Generating Sequences with Recurrent Neural Networks"
10. Graves and Jaitly (2014): "Towards End-to-End Speech Recognition with Recurrent Neural Networks"
11. Devlin et al. (2014): "Fast and Robust Neural Network Joint Models for Statistical Machine Translation"

[3] Paper ids generally have the form *XYY-ZZZZ*, where X is a single letter identifier (P is the main ACL, D is for EMNLP, Q is for TACL, etc.), *YY* are the final two digits of the year (2018 $\rightarrow$ 18), and *ZZZZ* is a per-proceedings identifier.