Project Ideas:

* Use Kumar model and test its effect on conceptual simplification.
* Try to get Newsela dataset.
* Try to access CoCo server for evaluation - or find a different method/metric
* Look at TextEvaluator for ideas - num of words before the first verb?
  + Make new function for evaluation/metric
* Apply model to different types of text
  + Look at how well it generalizes across domains
  + Scrape from Gutenberg
  + Use Arxiv
  + News articles
  + Abridged classics
  + https://github.com/SIMPATICOProject/simpa
* Manually create test dataset from a few abridged novels and compare against model’s test set (http://www.englishliteratureebooks.com/classicnovelsabridged.html)
  + Each take 5 texts and extract 5 examples of regular-simplified aligned sentences
    - Brandon take first 5
    - Melissa take second 5
* Look at which simplification methods transfer best - lexical, syntactic, conceptual, etc.
  + Try different weighting schemes when evaluating sentence probability?
* Explore Kumar code and model
* Add sentence normalization (to fix tokenization errors)

Notes:

* Model probably doesn’t know how to handle regional dialects like in some literature
* May not handle dialogue well

1. Transcribing/labeling stuttering events in stuttered speech
   1. 2021 state of the field: <https://arxiv.org/pdf/2107.04057.pdf>
      1. Some report detection accuracies in the 90s - one reported 99% accuracy
      2. Domain adaptation suggested as area for research - find more general features of stuttering
      3. Transformers have not been tried(much?) yet - attention mechanism could help
2. Text simplification
   1. Evaluate existing systems for performance on literature
   2. Domain adaptation - fine-tune existing system on literature
   3. Paragraph/document-level text simplification
   4. Unsupervised model training
      1. Constrain/penalize output based on readability, train for semantic consistency - may already be researched
         1. Devaraj, Ashwin, et al. (2021) “Paragraph-Level Simplification of Medical Texts.” appears to use constraints like this
   5. Consult ETS TextEvaluator for evaluation/constraint ideas
   6. I really like the Grammarly TST approach - we could build off of it (although we may not be able to test it like they did.. 50 epochs of fine-tuning would probably take way too long unless we get access to some very nice GPUs :) )
      1. Add model wrapper that lets the user specify which reading level the output should produce. The model could evaluate readability after each iteration until the target is reached.
      2. Test the model to discover what range of readability levels it can produce. This may be hard without reference data to ensure that the outputs are good quality.
      3. Modify the two FFNs that currently act as a label generator and a mask generator for the labels. Not that simple is bad, but since the setup is quite simple, maybe we could find a different architecture that improves performance.
   7. Awesome CoCo tool (conceptual complexity analyzer)
      1. Maybe we could build an application using this, or use its output as a constraint for a different text simplification model
   8. Build TS aligned corpus? Some potential abridged works we could scrape here and attempt to align with their full-length versions: <http://www.englishliteratureebooks.com/classicnovelsabridged.html>
3. Train ML model to recognize math operations entailed in word problems
4. Book recommendation system

Action items:

* Brandon to email Prof Bangalore about highlighted topics and seek advice
* Both: Find more papers on the highlighted topics

**Stuttering events**

Barrett, L., Hu, J., & Howell, P. (2022). [Systematic Review of Machine Learning Approaches for Detecting Developmental Stuttering.](https://discovery.ucl.ac.uk/id/eprint/10143157/1/Howell_T-ASL-08648_with_tables_and_figures.pdf) *IEEE/ACM Transactions on Audio, Speech and Language, Processing*.

Sheikh, S. A., Sahidullah, M., Hirsch, F., & Ouni, S. (2021). [Machine Learning for Stuttering Identification: Review, Challenges & Future Directions.](https://arxiv.org/pdf/2107.04057.pdf) *arXiv preprint arXiv:2107.04057*.

Alharbi, S., Hasan, M., Simons, A. J., Brumfitt, S., & Green, P. (2018, September). [A lightly supervised approach to detect stuttering in children's speech](https://eprints.whiterose.ac.uk/137999/1/2155.pdf). In *Proceedings of Interspeech 2018* (pp. 3433-3437). ISCA.

Alharbi, S., Hasan, M., Simons, A. J., Brumfitt, S., & Green, P. (2020). [Sequence labeling to detect stuttering events in read speech.](https://pdf.sciencedirectassets.com/272453/1-s2.0-S0885230819X0008X/1-s2.0-S0885230819302967/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjELn%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJIMEYCIQDcIh94lfcQOypkEaoKlA9HAB%2B3sXvQl82Yxl1i9vCE9gIhAMSySCx6By4DCI3rie5pdF8qXqTDNRsUBlNnNQ78ZrUAKvoDCCEQBBoMMDU5MDAzNTQ2ODY1IgzodBVlBixb3VX%2Fjq4q1wOnlS4OChbUg1CkNm20ERFikJJ4QP5jhuPNdkgejIGUd7qsEEwl0MENfMMZmdxQU9qwI56vK84ixUgYIcg7yWqN3p12IAPjKP1UX4QVd4BVw53nCd9m4SHdF%2BR17Z329FLAhR8qDkAWuklqoMuejOld7BCbFdU%2BUDFLwn18BpSBJS3pfC7bh%2BV%2BuHqPEYHX6wodoBZdNZY42oc3gKKmrpUBEHI5cRqTh0DxQfvXRMTnvfMqroTXUHseS%2FDy%2F4AVlIG%2FHSET%2FyvkwFNUtGaqnWk4HJfJOJ25TzcKKxazm%2BupRLXcsaN3ve5f%2FVVE51xhntWuoIZsRICTFDhQzEdJdGeDQp2NFLmcySMPv9jQR0K6%2Fuz6zBJFZQoWUT729egbS53dj5cXrcbu9WmU%2BWl%2Bt%2Bl74vF2LOHaWkS7eWHgkbstrDHGBko0d%2BG0qNyW4kEMENGh6cb9Dqh8U9EB%2BUTEOeTjxujgCn7UOro5e2GgEaqhQRqg3Pz1i12api0STUocd9vkJIMo81%2Bv4ptKyX1TKhGzZQj%2FzyEr0Q80nw%2FhTC9bHIlmoJbyRIAWmmhXAvzr6MU5JKL94EO2nwy07z4g%2BoaWi2uNeEW76R6afxpL7%2Blkyl34bE8bgV0wxe%2F6kAY6pAHso5%2BuwuMpCkvQxSShv55hjpSglQ%2F99VEkgqswtbSXN6GPMOHCwzHrxLYfkeQUbCEcwj%2Fk2n%2FLqWkTpaVa4d51r4LkFyzIgpKVi%2FcxJVnpIP1yEjq%2FYSgmcIZ4VF%2FIb%2BuQqlrgfUgnBoyiC8lPExs%2FR0so49ImlFIGDoCQ8hBUzOwSEJZH9qvj920NQpRpcF1tjIlAQKwRM0QoC1D1xt5t66T7og%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20220302T011904Z&X-Amz-SignedHeaders=host&X-Amz-Expires=300&X-Amz-Credential=ASIAQ3PHCVTYVV2VOX6U%2F20220302%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Signature=774f8a989f12256a44fc939ab9a5f1cc2783f5275ee0001f17a201eab4422253&hash=4306b9f1fcb2f38e74c24660571554663eca270fb8541babf1c2f0d80e26bec9&host=68042c943591013ac2b2430a89b270f6af2c76d8dfd086a07176afe7c76c2c61&pii=S0885230819302967&tid=spdf-88ef5eb2-fe70-4ee2-ac07-f219340943fb&sid=0fbbfeb548e5e641144927959bc0662a49e0gxrqa&type=client&ua=4c075a5456065a545603&rr=6e564d515f6d8c53) *Computer Speech & Language*, *62*, 101052.

<https://arxiv.org/pdf/2102.12394.pdf> SEP-28k

**Text Simplification -** [Overview of the SOTA](http://nlpprogress.com/english/simplification.html)

* [**Stajner et al. (2020) CoCo: A Tool for Automatically Assessing Conceptual Complexity of Texts**](https://aclanthology.org/2020.lrec-1.887.pdf) **automatic estimation of complexity**
  + Super cool neuroscientific approach
  + GitHub repo: ​​<https://github.com/ioanahulpus/cocospa>
* [**Omelianchuk et al. (2021) Text Simplification by Tagging**](https://aclanthology.org/2021.bea-1.2.pdf) **semi-supervised, used pretrained LM and data augmentation**
  + Not sure how the ensemble training works - it sounds like they’re training the models on data produced by the same models
  + Inference tweaks also weren’t very clear to me
* [**Surya et al. (2019) Unsupervised Neural Text Simplification**](https://arxiv.org/pdf/1810.07931)
  + 1 denoising auto-encoder + 2 decoders - unsupervised and semi-supervised approaches
  + Learns via a discriminator and classifier similar to GANs
  + Not clear how the discriminator and classifier work here
  + Evaluate with SARI, BLEU, F(R)E, Word Difference
  + Used human ratings for simpleness, fluency, relatedness
* [**Zhau et al. (2020) Semi-Supervised Text Simplification with Back-Translation and Asymmetric Denoising Autoencoders**](https://ojs.aaai.org/index.php/AAAI/article/view/6515/6371)
  + First they train a denoising auto-encoder and then use back-translation method to iteratively improve the “translation” decoders
  + Makes use of Simple PPDB, a dataset of simplification rules and associated probs.
  + Optimize outputs based on R, a harmonic mean of fluency (syntax), relevancy (semantics), and complexity (FK) scores
  + Use cross-entropy loss and REINFORCE backpropagation algorithm for R
  + Final loss is a weighted sum of the two
* [**Kumar et al. (2020) Iterative Edit-Based Unsupervised Sentence Simplification**](https://arxiv.org/pdf/2006.09639)**: edit-based simplification**
  + Code available
  + **This paper seems to have all the bells and whistles! It seems like they thought of everything… Maybe we could use their implementation and just try modifying parts of it?**
  + Generate candidate sentences through automatic parsing and subsequently performing common simplifying edits.
  + Scoring function is a product of fluency (using SLOR), simplicity (sent-length, FRE), and meaning preservation (cos-sim, entity tracking)
  + By adjusting thresholds for each score component, different aspects of simplification can be (de)emphasized (lexical, syntactic simplification, content reduction)
  + Use a GRU-RNN model
* [**Martin et al. (2020) Controllable Sentence Simplification**](https://arxiv.org/pdf/1910.02677.pdf)**: Controllable sentence simplification using sentence length/lexical complexity**
  + [**http://114.215.220.151:8000/20200504/Multilingual%20Unsupervised%20Sentence%20Simplification.pdf**](http://114.215.220.151:8000/20200504/Multilingual%20Unsupervised%20Sentence%20Simplification.pdf)
* [**Devaraj et al. (2021) Paragraph-level Simplification of Medical Texts**](https://arxiv.org/pdf/2104.05767) **(Devaraj paper you mention above)**
* [**https://www.frontiersin.org/articles/10.3389/fpsyg.2022.707630/full**](https://www.frontiersin.org/articles/10.3389/fpsyg.2022.707630/full) **(Talks about creating TS corpora**
* **Facebook’s** [**Martin et al. (2021) MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases**](https://arxiv.org/abs/2005.00352)
  + Github: <https://github.com/facebookresearch/muss>
* [**Maddela et al. (2020) Controllable Text Simplification with Explicit Paraphrasing**](https://arxiv.org/pdf/2010.11004.pdf)
  + Github: <https://github.com/mounicam/controllable_simplification>

1. Automatic text simplification
   1. What it is; how it is generally achieved; its applications; some history
      1. The goal of automatic text simplification is to make texts more readable to audiences that are challenged.
         1. Opt: Text simplification may also prove a useful preprocessing step for computer programs designed to carry out any number of NLP/NLU tasks (few sources in [Alva-Manchego, et al., 2020](https://aclanthology.org/2020.acl-main.424/)).
      2. Text simplification can have numerous applications. It can support numerous populations, including non-native English readers and those with low literacy levels, learning disabilities (e.g., autism, ADHD, and dyslexia), acquired language disorders (e.g., aphasia), and other cognitive impairments.
         1. The ability to read simplified texts can improve the quality of education and overall quality of life for these individuals. (find some sort of citation)
         2. It can help these populations to learn and gather information that may be otherwise inaccessible, e.g., official documents and news articles that have a significant impact on their lives
         3. It can help these populations participate in activities that others enjoy or that they previously enjoyed (e.g., reading for pleasure– aphasia book clubs use simpler texts so people can engage in conversation about books)
      3. Automatic TS is typically accomplished by reducing lexical complexity (i.e., using simpler words) and syntactic complexity (i.e., using simpler sentence structure) at the sentence level.
         1. Operations: simplify words, split sentences, reorder or remove phrases… (<https://www.frontiersin.org/articles/10.3389/fpsyg.2022.707630/full>)
         2. Beyond this, text simplification can also provide scaffolding (e.g., include a complex word followed by a definition written more simply)
         3. Text simplification may also include higher-level compression at the document level
   2. Challenges: data availability, meaning preservation, controllability, defining “simplification” and evaluation
      1. Data availability
         1. There are two primary parallel corpora used for the task of text simplification.
            1. Alignments of the typical English Wikipedia page and the Simple English Wikipedia page.

Many papers use Wikilarge, [**Zhang and Lapata (2017)**](http://aclweb.org/anthology/D17-1062), which is freely available and combines alignments shared by previous authors. There is a smaller dataset known as Wikismall.

“The dataset contains 8 (reference) simplifications for 2,359 sentences partitioned into 2,000 for development and 359 for testing. After removing duplicates and sentences in development and test sets, the resulting training set contains 296,402 sentence pairs.”

Wikilarge, assembled by Zhang and Lapata \citeyearpar{zhang2017sentence} and incorporating the work of previous researchers, is a freely available corpus built from aligned sentences in English Wikipedia and Simple English Wikipedia. The corpus contains nearly 300,000 sentence pairs, providing substantial data for machine learning approaches, however the data suffers from misalignments and often clumsily written simplifications (CITE).

* + - * 1. Newsela corpus ([Xu et al. 2015](https://aclanthology.org/Q15-1021/))

Zhang and Lapata (2017) also shared an alignment of this corpus

“ argue that Wikipediabased resources are suboptimal due to the automatic sentence alignment which unavoidably introduces errors, and their uniform writing style which leads to systems that generalize poorly. Newsela2 consists of 1,130 news articles, each rewritten four times by professional editors for children at different grade levels (0 is the most complex level and 4 is simplest). Xu et al. (2015b) provide multiple aligned complex-simple pairs within each article. We removed sentence pairs corresponding to levels 0–1, 1–2, and 2–3, since they were too similar to each other. The first 1,070 documents were used for training (94,208 sentence pairs), the next 30 documents for development (1,129 sentence pairs) and the last 30 documents for testing (1,076 sentence pairs). “

* + - 1. There are numerous other data sets that are not as large and are used less frequently as work has moved toward more data-hungry techniques.
         1. List here: <https://aclanthology.org/2021.findings-acl.233.pdf>
      2. Most resources are geared toward informational texts
    1. Defining simplification
       1. As mentioned earlier, text simplification has typically focused on reducing lexical and syntactic complexity
       2. Meaning preservation (to discuss below)
       3. Where is the line between text simplification and summarization?
    2. Meaning preservation
       1. Meaning preservation is a challenge in many natural language generation tasks (e.g., summarization, simplification…)
       2. Simplification should try to retain the meaning of the original text, but there may be an acceptable threshold for information loss
          1. E.g., removing descriptive details that are not essential may be OK even if some information is sacrificed
       3. Lexical simplification: though words may be in close proximity to others, there may be nuanced definitions, and collocations may need to be considered carefully (not just a meaning change but also a naturalness issue)
       4. Syntactic simplification: The original relationships between arguments may not be preserved when certain transformations occur

1. Previous work **(some summaries above, and the papers also list them out, MUSS being the most recent)**
   1. Manual text simplification
      1. ([Petersen & Ostendorf, 2008](https://www.isca-speech.org/archive_open/archive_papers/slate_2007/sle7_069.pdf) describes patterns in manual text simplification)
   2. Supervised methods
      1. ACCESS ([Martin et al., 2019](https://arxiv.org/abs/1910.02677))
         1. ACCESS is a sequence-to-sequence approach that utilizes a relatively simple method for learning constraints during training. Rather than imposing hard constraints during the text generation stage, specific well-defined strings (“control tokens”) are prepended to each input sentence, indicating the types and degrees of simplification for that particular training example.
            1. Number of characters (similar to words/sentence)
            2. Character-level similarity (proxy for rewording/paraphrasing)
            3. WordRank (proxy for lexical simplification)
            4. Dependency tree depth (proxy for syntactic simplification)
      2. MUSS+Wikilarge
   3. Unsupervised methods
      1. Kumar
      2. MUSS
         1. MUSS is an extension and only slight alteration of the ACCESS approach to training a sentence simplification model (Martin et al., 2021). Rather than training on a corpus of human-generated simplified texts, efficient search algorithms are employed in the Common Crawl corpus to find similar sentences based on cosine similarity. These sentence pairs are then prepended with descriptive control tokens similar to those used in ACCESS in order to train the model (pre-trained BART model in this instance).
2. More in-depth summary of Kumar 2020? ACCESS? MUSS?
3. Aim of project:
   1. Build a small parallel corpus of narrative texts
      1. We sought to evaluate existing TS systems on sentences from a different domain than those typically used. Rather than continue with news or informational content, we selected literature, specifically classic novels, for which abridgments have been published.
   2. Compare text simplification systems using EASSE tool ([Alva-Manchego et al., 2019](https://www.aclweb.org/anthology/D19-3009)). A meta-evaluation of TS metrics conducted by Alva-Manchego et al. ([2021](https://watermark.silverchair.com/coli_a_00418.pdf?token=AQECAHi208BE49Ooan9kkhW_Ercy7Dm3ZL_9Cf3qfKAc485ysgAAAr8wggK7BgkqhkiG9w0BBwagggKsMIICqAIBADCCAqEGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMV8GY1eNeNb09qG0hAgEQgIICct3kNd-SJmdfR_X-6PkF9tZ9OkFDFwm1OfByjb7n4QM8gSj0vTb57AbJ0k_QtBPOPQiWkHHZ3HmKm6zEAO7PNaKuOGGKLMHvzgUHI_omaXc48CPqRu5kxRKPhwADq7N_eNvXDbUYpjgcxDxwptvCmKpn1FFmN0a4BjalTlaLOjZUUYXgKmbbLKGUA8Sf-m8o8iFLrJS6vc4roHNPZMLeLBdjBZJe-L5RYAhjiH1dIhHQDRDNnEa-Bli47_sDElufR-QSZy6I61XXICx9pJuxLZlx_cu7YLMZjSmmgteRqygC6-gT3G6werZTooxCME3pP0i8ZijNSvLFV295Ad2-U0Te5tMrUaPTbZo0WkADS5xsZur7N9BJTIpWsTgCI-nrXfkIQPcBF-YO0U431mLlLBCMwb6vHKJNxMp2Qr5pUqn4uU-Z0rUrcs-RdLgMQg162Ofv6QPd9K-FJJQLv60uRWe5nzPj3s9GZDJ_ChNPHbsddDReYFeeTm75Mz9J7v3IGjl3R1ZaHNM2EGDDpRQQoqSZ-4VZLWXH-IaxeoEe3UbJgzbLAH35hhx2fGVKfXw6u2WChD69MRW4VTBI81NFLtqNtzvp2CRA6JtiMIsuiZRDFAo8105a_v5dIiEKTxOkqFVMWp42VjIDNiZdu5Q3dIQ6nBUWZr0T2wg2gQJlHBuLMw49SXVhTKIU4lZ5TmB6Ssif-T3YgX682WKK4Z8YMRihaeGlK63w6S9DrhEBLV9ydn0xl02yznn-t436lALNfM4csyAtU312MbRBc3-wUTwYzGZj9nd2K0PNy3K9OzUixO53SZKaA1XO8uKRggW7jziE)) maintains that no one metric should be considered in isolation, but that an ensemble of the existing metrics are needed to approximate human judgment.
      1. Quality of the results
         1. SARI when compared to human reference ([Xu et al., 2016](https://aclanthology.org/Q16-1029.pdf))
            1. SARI ([Xu et al., 2016](https://aclanthology.org/Q16-1029.pdf)) is a metric developed specifically for evaluating text simplification models and, as such, is considered first when ranking different models/outputs.
            2. As stated in its name, the SARI method compares System output Against (multiple) References *and* against the Input sentence. Each SARI score considers the F1 scores of proper n-gram additions, deletions, and constants. The arithmetic mean of these F1 scores is then taken as the text’s SARI score.
         2. FKGL
            1. Flesch-Kincaid Grade Level is a simple algorithm for estimating which reading level a text is best suited for (Kincaid et al., 1975). It is a linear combination of two variables, the number of words per sentence and the number of syllables per word, such that the simpler a text is, the lower its FKGL.
         3. BLEU
            1. The Bilingual Evaluation Understudy (BLEU) method was developed for assessment of machine translations, but is a common reference for various sequence-to-sequence NLP tasks, owing to its frequent correlation with grammaticality and meaning preservation ([Papineni et al., 2002](https://aclanthology.org/P02-1040.pdf), [Stajner et al., 2014](https://aclanthology.org/W14-1201.pdf)).
            2. BLEU scores should not be relied upon as a primary metric, because they do not handle various simplification edits well, and they have been observed to correlate inversely with sentence simplicity ([Sulem et al., 2018](http://aclanthology.lst.uni-saarland.de/D18-1081.pdf)). Several equally simple and grammatical model outputs may receive starkly different BLEU scores due to varying amounts of deviation from the reference target. BLEU scores may, however, still serve as a useful supplement to assess closeness of match to a specific simplification target.
         4. BERTScorePrecision? (recommended in [Alva-Manchego, 2021](https://watermark.silverchair.com/coli_a_00418.pdf?token=AQECAHi208BE49Ooan9kkhW_Ercy7Dm3ZL_9Cf3qfKAc485ysgAAAr8wggK7BgkqhkiG9w0BBwagggKsMIICqAIBADCCAqEGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMV8GY1eNeNb09qG0hAgEQgIICct3kNd-SJmdfR_X-6PkF9tZ9OkFDFwm1OfByjb7n4QM8gSj0vTb57AbJ0k_QtBPOPQiWkHHZ3HmKm6zEAO7PNaKuOGGKLMHvzgUHI_omaXc48CPqRu5kxRKPhwADq7N_eNvXDbUYpjgcxDxwptvCmKpn1FFmN0a4BjalTlaLOjZUUYXgKmbbLKGUA8Sf-m8o8iFLrJS6vc4roHNPZMLeLBdjBZJe-L5RYAhjiH1dIhHQDRDNnEa-Bli47_sDElufR-QSZy6I61XXICx9pJuxLZlx_cu7YLMZjSmmgteRqygC6-gT3G6werZTooxCME3pP0i8ZijNSvLFV295Ad2-U0Te5tMrUaPTbZo0WkADS5xsZur7N9BJTIpWsTgCI-nrXfkIQPcBF-YO0U431mLlLBCMwb6vHKJNxMp2Qr5pUqn4uU-Z0rUrcs-RdLgMQg162Ofv6QPd9K-FJJQLv60uRWe5nzPj3s9GZDJ_ChNPHbsddDReYFeeTm75Mz9J7v3IGjl3R1ZaHNM2EGDDpRQQoqSZ-4VZLWXH-IaxeoEe3UbJgzbLAH35hhx2fGVKfXw6u2WChD69MRW4VTBI81NFLtqNtzvp2CRA6JtiMIsuiZRDFAo8105a_v5dIiEKTxOkqFVMWp42VjIDNiZdu5Q3dIQ6nBUWZr0T2wg2gQJlHBuLMw49SXVhTKIU4lZ5TmB6Ssif-T3YgX682WKK4Z8YMRihaeGlK63w6S9DrhEBLV9ydn0xl02yznn-t436lALNfM4csyAtU312MbRBc3-wUTwYzGZj9nd2K0PNy3K9OzUixO53SZKaA1XO8uKRggW7jziE))
            1. Essentially the average token cosine similarity score ([Zhang, 2020](https://arxiv.org/pdf/1904.09675.pdf)). Averages across each token in the output with its most similar token in the reference. Also, I think, weights each token by idf, so that rarer tokens (in corpus) are more influential in sentence score.
         5. Cosine similarity?
            1. Perhaps more holistic than BERTScore, since it considers sentence embedding rather than contextualized token embeddings. Doesn’t give precedence to rare words though.
         6. Model trained on Corpus of Linguistic Acceptability (COLA)?
      2. Time efficiency?
      3. Controllability
4. Method
   1. Data we created
      1. Set of 100 sentences from freely available Project Gutenberg
         1. Some pulled from abridged versions <http://www.englishliteratureebooks.com/classicnovelsabridged.html>
         2. Some manually simplified from most popular Project Gutenberg texts
   2. What we did
   3. Charts?
5. Discussion
   1. Unsupervised+Wikilarge seems to demonstrate promising results on data from a different domain than WikiLarge
   2. Unsupervised alone may not yield the most simplified results but rather just paraphrased results
   3. Controllability as it is used now may have some limits/needs to be tested more to ID what is most helpful
      1. Is shorter really better?
      2. Types of simplification may depend on the context
         1. Reading for pleasure: OK if more complex vocab is simplified
         2. Reading to learn: Might need to retain harder vocab but provide scaffolding, e.g., define terms
6. Conclusion

Goal for Tuesday:

* Brandon: Unsupervised system classic lit
  + Reorder leaves: False ?
* Brandon installs ACCESS
* Both: Run classic lit complex data using ACCESS
  + NbChars\_1.00, NbChars\_0.75, NbChars\_0.50, NbChars\_0.25
  + NbChars\_1.00 + [LevSim\_\*\*, WordRank\_\*\*, DepTreeDepth\_\*\*]
* First section of paper (Melissa starts on Friday)
* Melissa: Look into other data sets
* Melissa: Think about how we should evaluate
  + BLEU/SARI (single reference)
  + Human ranking
  + Other automated metrics: Grammaticality? Fluency? Adequacy/semantic similarity?
* Brandon: Rewrite some sentences from Project Gutenberg lit
* Melissa: Rerun all models and run human-matched parameter versions with 0.95 & 1.00 word rank ratios, change other parameters to nearest 0.05
* Brandon: Run sentence-by-sentence cosine similarity over output predictions (and also average across all sentences)
* Brandon: Run EASSE to collect sentence-by-sentence scores
* Melissa: Send rough descriptions of rating scales for Adequacy, Simplicity, Fluency
* Brandon will finish annotating by Wednesday
* Brandon - export individual sentence results to CSV by Friday
* Fill in all sections except for results and discussion by Saturday - can start them too.
* Multiple sections are named Evaluation, Automated Metrics, and Human Evaluation. Can we reorganize or rename these?
* Add heat map, 1-2 box plots?
  + 1 box plot with all wikilarge vs human comparisons - show how similar they are
* Talk about how well MUSS and access transfer to our data set
* Bertscore precision and recall track with compression ratio, seems like not much information is lost when the sentences are compressed more.
* BERTscore tracks best with human judgment of quality, fluency. Due to use of embeddings rather than n-grams probably

Old: <https://www.overleaf.com/project/6269dab63c70825d50d5c013>

New: <https://www.overleaf.com/2917386673hkkmqqwvnftm>

<https://colab.research.google.com/drive/1j2zW6dZn0qJnHUQrJjV9ASlMiUgbQQC3?usp=sharing>

<https://docs.google.com/spreadsheets/d/1GUIpLans2j70dzWzpsTN9B1l36heuDk1w2QVk_rCGu0/edit#gid=0>