word2vec: We will soon see how to learn vectorial representations for words using the bag-of words model, skip-gram model and the Glove model. Your task in this assignment is to train such word representations for non-English languages (specifically your native language).

- Crawl pages from the web in your native language (pick Hindi only if both the members of the group have Hindi as the native language, otherwise pick a non-Hindi language). The more data you crawl the better will be your word representations (**we are releasing this assignment early so that you can start crawling data now itself even before we cover word2vec in class**).
- In your report, list down all the sources from which you have crawled data (the URLs for these sources).
- Clean the above corpus (remove html tags, special characters, etc.) and break it into sentences. You should store this corpus in one or more text files such that each text file contains one sentence per line.
- Use existing code for the 3 models mentioned above (b-o-w, skip-gram and Glove) from online sources and train word representations of different sizes. In the report, mention how you experimented with different hyper-parameters (learning rate, batch size, embedding size, etc) and the observations from these experiments (for example, what happens when you increase the amount of training data). Just to be clear, you don't need to code these models but just find existing code and experiment with it (1 bonus mark to the group which is the first one to identify an online code and post a link on moodle). **You can start getting familiar with this code, how to run it, etc. even before we cover word2vec in class**.
- Towards the end of the lecture we will see different ways of evaluating word representations. Use your ingenuity to come up with a test benchmark which allows you to compare the different methods as well as different embedding sizes (100, 200, 300, etc). For example, you can design a small test set which allows you to evaluate semantic relatedness, synonymy detection or word analogy in your native language. Even a small benchmark consisting of 25-30 test instances is fine. You can refer to the benchmarks mentioned in `https://transacl.org/ojs/index.php/tacl/article/view/570/124` to get an idea of the evaluation.

Note:

- You will be able to understand and appreciate this assignment fully only after we cover word2vec in class. However, there are certain very simple mechanical/programming jobs which need to be done as a part of the assignment (crawling/cleaning pages, downloading and setting up the code). We felt it would be best to use your time to finish off these mechanical tasks before we cover word2vec in class. **We encourage you to mutually share data with other groups.**

- One of the intentions of this assignment is to give you a flavor of how one goes about solving problems in real world situations. The data is almost never served on a platter but you need to find smart ways of gathering data. Second, in most research problems you start by using an off-the-shelf solution for the problem before thinking about fresh solutions. Third, in most real-world problems you will have to make a small evaluation benchmark consisting of a varied test set which largely covers the challenges involved in the problem. Finally, you need to make certain observations about the behavior of these algorithms on your data and then suggest improvements to address these limitations (of course, we do not expect you to propose anything new but we would like to read about your observations).