### Description:

Word embeddings are a key component of language modelling. They map words to vectors of real numbers. Word embeddings attempt to capture semantic similarities between words by judiciously positioning them within a higher dimensional space. Words that are close within that space should be semantically related. There are many strategies for learning word embeddings from text data.

We are interested in word embeddings for clinical language. Clinical language is rich in medical terms that are poorly covered by general-purpose English language embeddings. This limits our ability to use pretrained general-purpose word embeddings.

While it is possible to learn word embeddings for the clinical domain, it could be hard to evaluate them. As NLP engineers we would like to evaluate them on a series of conditions we expect the embeddings would satisfy. As such, a framework that could evaluate and compare produced embeddings could be of great of use. While the evaluations may not be comprehensive, they would provide a minimal level of assurance.

### Framework Development:

In brief, the framework should load clinical word embeddings and run evaluation tests on them. Then produce the results in some human readable format.

* In software terms, the framework is similar to a test harness (A publicly available open-source test harness could be a good starting point for the development).
* However, unlike a test harness which runs software tests, the framework runs evaluations that quantify the quality of word embeddings.
* Ideally, the framework should be flexible, configurable and modular. We should be able to add tests to it. We should also be able to configure what tests to run.
* Unlike standard software tests which have pass/fail criteria, the evaluations would most likely produce evaluation metrics (numbers). These metrics should be reported.

A key purpose of the framework is to compare embeddings. Ideally the framework should support evaluating multiple word embeddings and present them in a way that facilitates comparison.

### Evaluation Tests:

Tests themselves should also be parametrized and configurable. The following are some of the tests that we would like to have.

* Vicinity analysis for synonymous concepts identified in the [Unified Medical Language System (UMLS).](https://secure-web.cisco.com/1EKkk8MNsYU68JHvJEWhjAX9VPLBjRnIO-peSM1bw9bYlTNXd9X9Yjt8oL7cURcupsv-8fxnX2fRSaJIwEJFXmev6W2TAqIysXg4BXF13we2pvXN-I5DA75o82Pws0Cl0bNOFC0Vzze6w5yY_j-CA4YWKtEWyXxdnW9xj9upl3fzukdr-U0M4EOHrG-KhYMCeUdL9OtRcGEGJsOX_M6gkQE5Pn6qyEdl2Pq2pj7qwCtcj1Nich9GiZBE-pVRFHX0_TcVqZcfI5gy_3-SGkKtuElZG1agetGSqwXxGALE21OCbKgo_dZ7gqCQ4BGheaqaX0GxXOKIbBd9K7wg_zCKcX-52uMcJfuR-ISfvnIikF_pobaHeFHe-ezbeO_1qmFAAXMtafgWqG8IQCShMHqEG-gSoJ6d2iZGob1W5-qGk4pvqT49lzQW8eIqJTDkBMb9ymIDeinj60w6vreGykDjYDQ/https%3A%2F%2Fwww.nlm.nih.gov%2Fresearch%2Fumls%2Findex.html) UMLS contains a metathesaurus that defines synonymous and related terms. A good embedding would have to place related terms close together. Based on UMLS data, one could develop a test that checks to what extent synonymous terms are close to each other in the learned embeddings.
* A sample ICD9 classification task (on [MIMIC](https://secure-web.cisco.com/16GHO73OUws-IT4joygvMRWXMHIgoq9KgzQSYmyFYVVCj5ddDVDouBhmyjFkKFseM-umSVhtdXOvlkZqKi3MYjNSvh_w9U1Ig-NjHNjkGXI5Su9qZ6wKoPXME4VH1_2hg9ToW6v-eT_znMPeDR2rK83l6gzxQKfOhVxhy5gJDqVhsJciqSIM_48Qt1SNCvjUqZSxf20uPIPVmQ_SS8QnphoEFjNIM7CyTL3J7Gux_VkS-4NxRdEdOL7zTaVi3j0ITMT_XDdiO6kRikwGHF4rjrx5GmgrVAwyiByiZm1dcZyD-hZTqd9TTFONK9RBtcA_UAImRHZ66dcsd_oLNhj9mTbH6y8P6YXq6DXiBZsZgD-N6isMzmlW9l0ErGxsGKwnVkS-IsbbXSGRn-DhMGwtJ1fuCPY-__HpDWhBdEvry90zx65upSfptPMBwNQPfFxGc/https%3A%2F%2Fmimic.physionet.org%2F) data). The documents within the MIMIC dataset are annotated for ICD9 codes. This task would involve developing a minimal but adequate machine learning task that tries to guess ICD9 codes from the medical text.
* Other tests if students wish to develop more

### Research Component:

The research component of this project involves learning clinical word embeddings using various strategies, and then using the framework above to compare them. Word embeddings can be learned from [MIMIC](https://secure-web.cisco.com/16GHO73OUws-IT4joygvMRWXMHIgoq9KgzQSYmyFYVVCj5ddDVDouBhmyjFkKFseM-umSVhtdXOvlkZqKi3MYjNSvh_w9U1Ig-NjHNjkGXI5Su9qZ6wKoPXME4VH1_2hg9ToW6v-eT_znMPeDR2rK83l6gzxQKfOhVxhy5gJDqVhsJciqSIM_48Qt1SNCvjUqZSxf20uPIPVmQ_SS8QnphoEFjNIM7CyTL3J7Gux_VkS-4NxRdEdOL7zTaVi3j0ITMT_XDdiO6kRikwGHF4rjrx5GmgrVAwyiByiZm1dcZyD-hZTqd9TTFONK9RBtcA_UAImRHZ66dcsd_oLNhj9mTbH6y8P6YXq6DXiBZsZgD-N6isMzmlW9l0ErGxsGKwnVkS-IsbbXSGRn-DhMGwtJ1fuCPY-__HpDWhBdEvry90zx65upSfptPMBwNQPfFxGc/https%3A%2F%2Fmimic.physionet.org%2F) train data. The following are some strategies to learn word embeddings:

* Word2vec
* GloVe
* Starspace
* Fasttext
* Others…
* While not exactly embeddings, BERT and XLNET and other encoders might also be analyzed using this tool

It is up to the students to experiment with some or all the embedding strategies above.

### Task Breakdown:

The following is a rough breakdown that illustrates one way to split the full project into tasks. Tasks 1-4 are either independent or loosely coupled and as such can be handled in parallel by different teams or team members. Task 5 involves tying all components together and as such is dependent on the other tasks.

1. Developing the framework
   1. Defining the interface for evaluation tests. The interface would encompass inputs, configs and outputs.
   2. Implementing the engine that would read the configs and run evaluations
   3. Implementing the result and report generating components
2. Vicinity Analysis based on UMLS (finalizing is dependant on 1.a, but bulk of the work can be done independently)
3. ICD9 classification task (finalizing is dependant on 1.a, but bulk of the work can be done independently)
4. Learning word embeddings from MIMIC data. This can be further broken down into independent tasks.
5. Put all the components together to evaluate learned word embeddings
   1. Align the learned word embeddings into a format the framework can read
   2. Create necessary configuration files
   3. Run the evaluation on learned embeddings
   4. Obtain, report and analyze result

### Desired Outcome

For participating students, this project offers the opportunity to apply several aspects of software development and machine learning, along with the potential of publishing results in a conference or journal. Almost all parts of the project support additions or modifications, and the students are welcome to add or remove from the project based on their personal interests or analyses.

For us, the research component might identify strategies that could be beneficial to us. The framework would allow us to evaluate and compare our embeddings. When the project is complete we would likely add further evaluation tests of our own targeting our internal projects.

### Sponsors:

Nasr Madi, Nuance Nasr.Madi@nuance.com

Weiyi Sun, Nuance [Weiyi.Sun@nuance.com](mailto:Weiyi.Sun@nuance.com)