**Report**

**Semantic Textual Similarity**

**Semantic Textual Similarity:**

It deals with determining how similar two pieces of texts are. This can take the form of assigning a score between 0 and 1. Related tasks are paraphrase or duplicate identification.

STS is the assessment of pairs of sentences according to their degree of semantic similarity. The task involves producing real-valued similarity scores for sentence pairs.

**Approach:**

Approach followed in this project is word2vec + cosine similarity to calculate distance between two words. Google’s **word2vec** is implemented by [Gensim](https://radimrehurek.com/gensim/) library in this project due to its easy yet powerful implementation.

**Word2vec:**

The **word2vec** tool takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The resulting word vector file can be used as features in many natural language processing and machine learning applications.

**Gensim:**

Gensim is billed as a Natural Language Processing package that does ‘Topic Modeling for Humans’. But it is practically much more than that. It is a leading and a state-of-the-art package for processing texts, working with word vector models (such as Word2Vec, FastText etc) and for building topic models.

**Why to use word2vec and Gensim package?**

In this project a variant of word2vec i.e doc2vec is used. Doc2vec converts text data in vectorized form to create word embedding using a shallow neural network model.

**Advantage of using doc2vec instead of traditional bag of words model:**

* Mathematical operations on words.
* Context preservation.
* Dense representation, hence low memory usage and efficient .

**Advantage of Gensim library:**

* Neural network implementation.
* Doc2Vec implementation.
* Reduces file size.
* Easy to use.
* Platform independent.
* Cleaner code.

**Steps Taken:**

1. Import Dataset : Load dataset in memory.
2. Familiarize with dataset : Know about data like number of rows/columns, type of data, missing values e.t.c.
3. Data clean up : Clean the data by imputing missing values, discarding irrelevant data, removing symbols and numbers, e.t.c
4. Data preparation : Prepare data in appropriate format before feeding it to the model.
5. Model : Create and train model with appropriate hyperparameter tuning.
6. Evaluate model : Evaluate model’s performance by various metrics.