**Pre trained Word2vec model:SO\_word2vec**

The software development process produces vast amounts of textual data expressed in natural language.

Outcomes from the natural language processing community have been adapted in software engineering research for leveraging this rich textual information; these include methods and readily available tools, often furnished with pre-trained models.

State of the art pre-trained models however, capture general, common sense knowledge, with limited value when it comes to handling data specific to a specialized domain.

There is currently a lack of domain-specific pre-trained models that would further enhance the processing of natural language artifacts related to software engineering.

To this end, this word2vec model is trained over 15GB of textual data from Stack Overflow posts.

**Pre-trained Model:**

Converted the whole text into lowercase in order to avoid information loss due to the informally written language of the resource, where different spellings and cases appear for referring to the same word.

After initial preprocessing, 15GB of clean text remained, amounting to over 11 billion words.

Further preprocessing involved removal of stop words, i.e., words that frequently appear in text and do not carry meaning associated with concrete concepts (typical examples of stop words include: and, or, it, the, is, are, if, a, me). After stop word removal, over 6 billion of words remained, which comprised the final data set that was fed into the training algorithm.

For training the model we used the open source Python library gensim.

Model training over 3 epochs, which took about 11 hours to complete and trained the word2vec model with vector dimensions of 200 features and the window for checking word co-occurrences set to 5.

Dimensionality is set lower than the 300-feature state of the art Google news vectors since for our much smaller dataset with limited vocabulary, a lower dimension should be sufficient for capturing the necessary features.

The resulting model consists of a vocabulary of 1,787,145 keywords.

The pre-trained model is stored in a .bin file (of approximate size 1.5 GB)

<http://doi.org/10.5281/zenodo.1199620>

**Loading the model**

from gensim.models.keyedvectors import KeyedVectors

word\_vect = KeyedVectors.load\_word2vec\_format("SO\_vectors\_200.bin", binary=True)

**Querying the model**

Examples of semantic similarity queries

words=['virus','java','mysql']

for w in words:

try:

print(word\_vect.most\_similar(w))

except KeyError as e:

print(e)

print(word\_vect.doesnt\_match("java c++ python bash".split()))

Examples of analogy queries

print(word\_vect.most\_similar(positive=['python', 'eclipse'], negative=['java']))

**Reference:**

<https://github.com/vefstathiou/SO_word2vec/blob/master/MSR18-w2v.pdf>

<https://github.com/vefstathiou/SO_word2vec>