

```
In [ ]: from gensim.models.doc2vec import Doc2Vec
import numpy, math

f='doc2vecR.200.30.20.5.1550908281.eAp.trained' # THE TRAINED DOC2VEC MODEL
mod = Doc2Vec.load (f)
```

```
In [ ]: # Define the Cosine Similarity Function
def cos_sim (av, bv):
    return (sum(av*bv)/math.sqrt(sum(av*av)*sum(bv*bv)))
```

APIs are the "words" in the model, rest are document tags -

Authors have a structure like : John Doe < johnd@me.com >

Projects have a structure like: GitHubUsername_ProjectName

Languages have no "<", "@", ">", or "_"

```
In [ ]: # Example - Most similar APIs to an author
mod.wv.most_similar(positive=[mod.docvecs['Ayende <Ayende@ayende.com>']])
```

```
In [ ]: # Example - Most similar APIs to a project
mod.wv.most_similar(positive=[mod.docvecs['07101994_SignalR']])
```

```
In [ ]: # Example - Most Similar APIs given other APIs
mod.wv.most_similar('numpy')
```

```
In [ ]: # Example - Similarity between API & Author
vec = mod.wv.get_vector('net.ravendb.abstractions.basic.Reference')
avec = mod.docvecs['Ayende <Ayende@ayende.com>']
print(cos_sim(vec, avec))
```

```
In [ ]: # Example - Similarity between API & Project
vec = mod.wv.get_vector('net.ravendb.abstractions.basic.Reference')
pvec = mod.docvecs['07101994_SignalR']
print(cos_sim(vec, pvec))
```

```
In [ ]: # Example - Similarity between Author & Project
avec = mod.docvecs['Ayende <Ayende@ayende.com>']
pvec = mod.docvecs['07101994_SignalR']
print(cos_sim(avec, pvec))
```

```
In [ ]: # Example - Doing a "King:Man::Queen:?" type of comparison
mod.docvecs.most_similar(positive = [mod.docvecs['Ayende <Ayende@ayende.com>']+
    mod.docvecs['0x0ade_tModLoader']-
    mod.docvecs['07101994_SignalR']])
```

```
In [ ]: # Example - Similar Tags (Tags include projects, authors, and language)
mod .docvecs.most_similar( [ 'Ayende <Ayende@ayende.com>' ])
```

```
In [ ]: # Example - Tag similar to multiple APIs
""" Use Case: A developer wants to know what are the best projects/which
authors have similar expertise given they are interested in some
specific APIs - e.g. they might want to work on a project that uses
```

```
gensim & numpy
Note - there is no way to request for a project tag or author tag
specifically, so you need to comb through the top results
(ideally using a for-loop) to find what you're looking for,
e.g. in this case, the project appears at the 98th position
"""
mod.docvecs.most_similar(positive = [mod.wv.get_vector('gensim')+
                                     mod.wv.get_vector('numpy')], topn = 100)
```