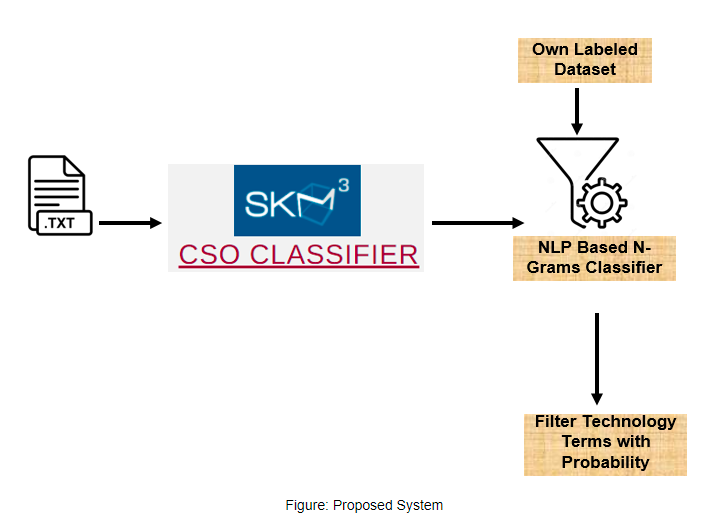
**Improving CSO performance**

**Eyong**: Technology detection is very contextual term actually.

* Let's say you system is dealing with software then terms Computer Programming, Compiler Construction, Machine Learning Algorithms etc are valid technology terms. In this case some terms such as Core i7 System, Read Only Memory etc. might not be relevant.
* Similarly if you are dealing with hardware, then some software terms may not be relevant.
* If you are talking about general computer science then there is requirement to look into your own definition of technology. For example the Core i7 System is the technology but marked by CSO classifier.

**My Solution/ Suggestions**: Use another filter over CSO Classifier. The output of CSO will be input to our filter classifier (based on N-Grams model - this classify terms on character level)

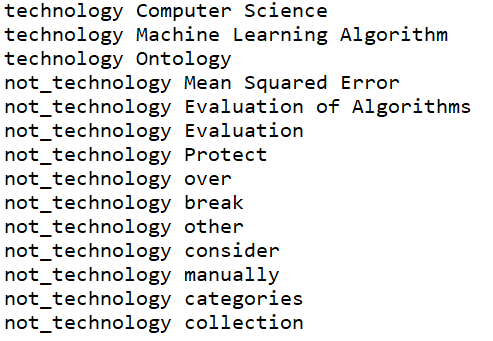


* Technology Identification is guaranteed, because CSO will extract technology terms, that will be later filtered by proposed system.
* Technology extraction will be extracted using CSO.
* Update New technology terms that are not part of CSO will be identified using our N-Gram Based Technique.
* Challenge: Producing our own list of technology terms (but this is not difficult issue)

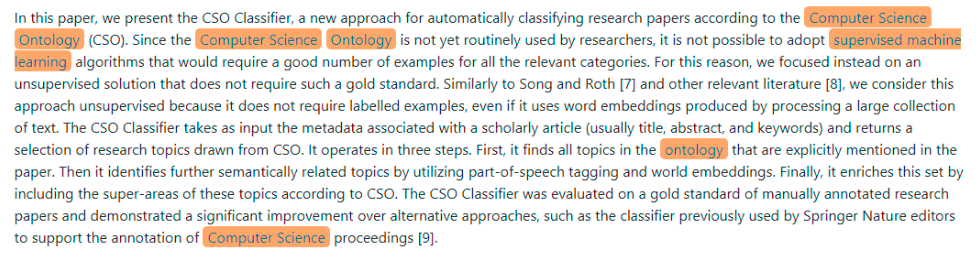
**Just for You**

(This is Just a Proof of Concept, it’s your choice if you want to show this to your company)

**Training Data**

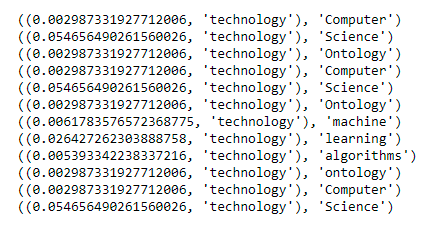


**Labelling by CSO Classifier**



**Filtering by Proposed Technique**

Each term (output by CSO classifier) will be filtered using own dataset and classifier which will calculate the probability and label of each term. This can filter out non technology terms of CSO output.

****

**Proposed Technique in Action**

Let say CSO classifier gives you list of terms which are detected as technology terms, but there are some terms as give “mean square error” which is not a technology, proposed technique will remove this or tag this not technology on the basis of its own dataset.

