Semantic Annotation of Documents

Team 20

Sharvil Katariya
Rohit SVK
Nikhil Chavanke

Mentor - Priya Radhakrishnan Course Instructor - Vasudev Varma

Problem Statement

- Semantic Annotation of documents - To annotate a document with a Wikipedia article that matches its contents most closely.
- Assigning wikipedia topics to any document using
  - Random forests
  - Gradient Boosting Classifier
  - Support Vector Machines
  - Logistic Regression
  - or any popular classification algorithm.

Methodology

- Data Pre - Processing.
- Training word and paragraph vectors.
- Training new vectors.
- Training the classifier using supervised learning.
- Testing the classifier

Pre - Processing

- This phase involved mapping the field of the Research Paper to a higher or more broader topic.
- The preprocessing phase also involved the
  - Removal of Stopwords.
  - Stripping of excess whitespaces.
  - Removing Punctuations.
  - Removing Tags from text, etc.
- Splitting the data into train, test texts.

Training word and paragraph vectors

- Word2vec representation is used to train words in the corpus
- Doc2vec representation is used to train paragraphs in the corpus
- Dimension, which can be tweaked, is set to 400
- Epochs can be set to 50 for deep learning. An epoch is just a measure of the number of times all of the training vectors are used once to update the weights.
- The abstract in the corpus is represented in the vector space model.

Training and testing the classifier

- Classifier takes list of arrays, that is computed model vectors, corresponding to its labels.
- The randomly split training and testing data, enables the classifier to use these model vectors for the subsequent training of the classifier.
- The model vectors of the testing data are sent to the classifier and compared with the labels associated with the testing data, with the help of various evaluation parameters.

Corpus Used

- The Dataset used is that of ACM research papers.
- Each Datapoint in the Dataset contains
  - Title
  - Abstract
  - Authors
  - Location
  - Timestamp
  - Conference
  - Index Number
- Number of Research Paper with abstract: 247543
- Test-Size: 10% of the entire dataset, which is split randomly.

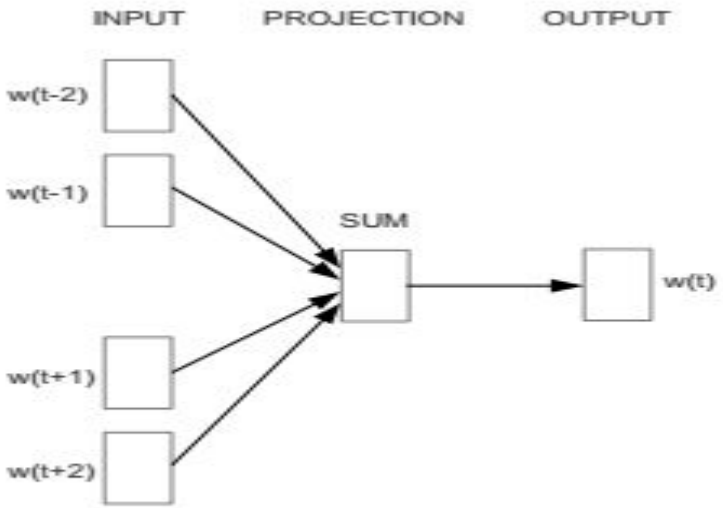
Feature Selection

- The dimensions of the vector representation of the paragraph is taken as the features of the data and trained.
- It is up to our novelty to set the no of features.
- More the number of features, the more is the result obtained on the trained data.
- However, one must ensure to avoid problems of Data Overfitting.

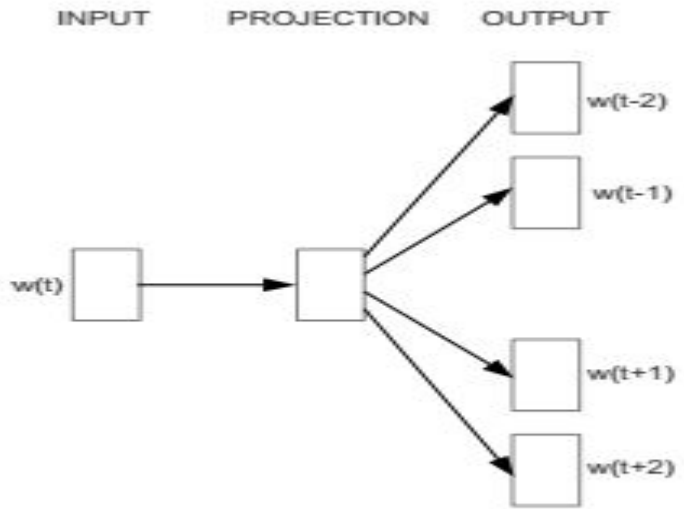
Architecture Models

- As stated in the paper, there are 2 architectures, continuous bag of words based (CBOW) and the other skip-gram (PV-DBOW) based
- Word2Vec and Doc2Vec are trained using individual and both architectures and the results are visualised.
- Doc2vec is similar to Word2Vec, except now we represent not only words, but entire sentences and documents.
- Doc2Vec enables us to represent an entire sentence using a fixed-length vector and proceeding to run all our standard classification algorithms.

Architecture for CBOW and Skip-gram method



CBOW

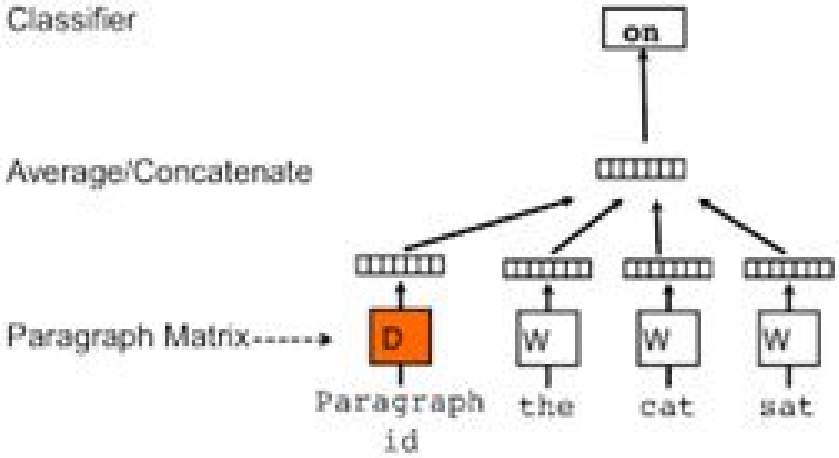


Skip-gram

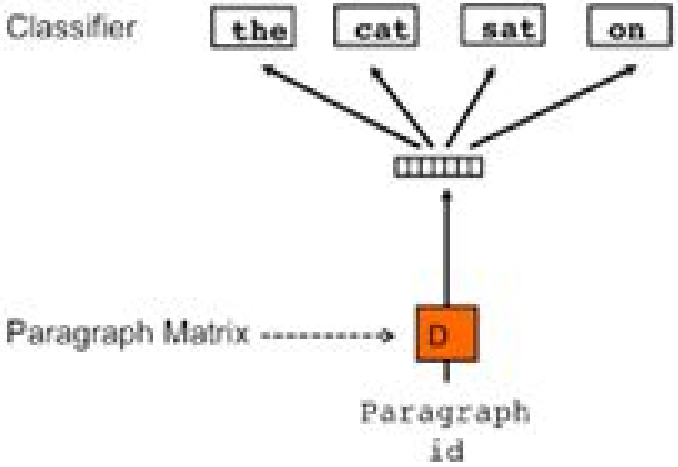
CBOW forces the neural network to predict current word with the help of surrounding words, and Skip-Gram forces the neural net to predict surrounding words of the current word.

Training is essentially a classic back-propagation method with a few optimization and approximation tricks (e.g. hierarchical softmax).

Architecture for Doc2vec



Distributed Memory (DM) model



Distributed Bag of Words (DBOW) model

Architecture for Doc2vec

- DM (Distributed Memory) attempts to predict a word given its previous words and a paragraph vector. Even though the context window moves across the text, the paragraph vector does not (hence distributed memory) and allows for some word-order to be captured.
- DBOW (Distributed Bag of Words) predicts a random group of words in a paragraph given only its paragraph vector

Evaluation Parameters

Mean Average Precision (MAP)

- Useful for multiple relevance.
- Mean average precision (MAP) is the Average of the average precision value (average of the precision values at the points at which each relevant document is retrieved) for a set of queries.
- If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero

Evaluation Parameters

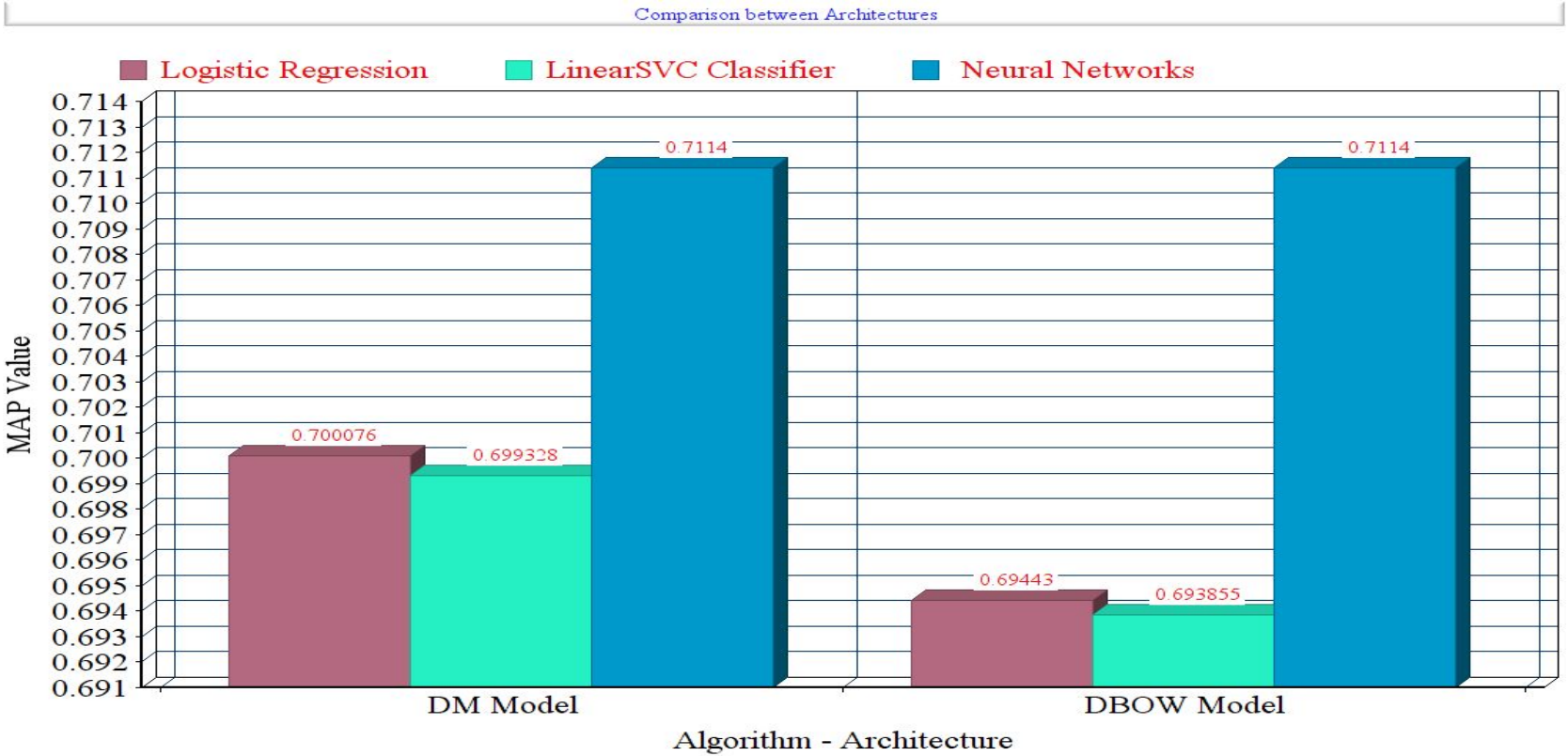
Normalized discounted cumulative gain (NDCG)

- NDCG measures the performance of a recommendation system based on the graded relevance of the recommended entities.
- Uses graded relevance as a measure of the usefulness, or gain, from examining a document.
- Gain is accumulated starting at the top of the ranking and may be reduced, or **discounted**, at lower ranks.
- Discount Function used

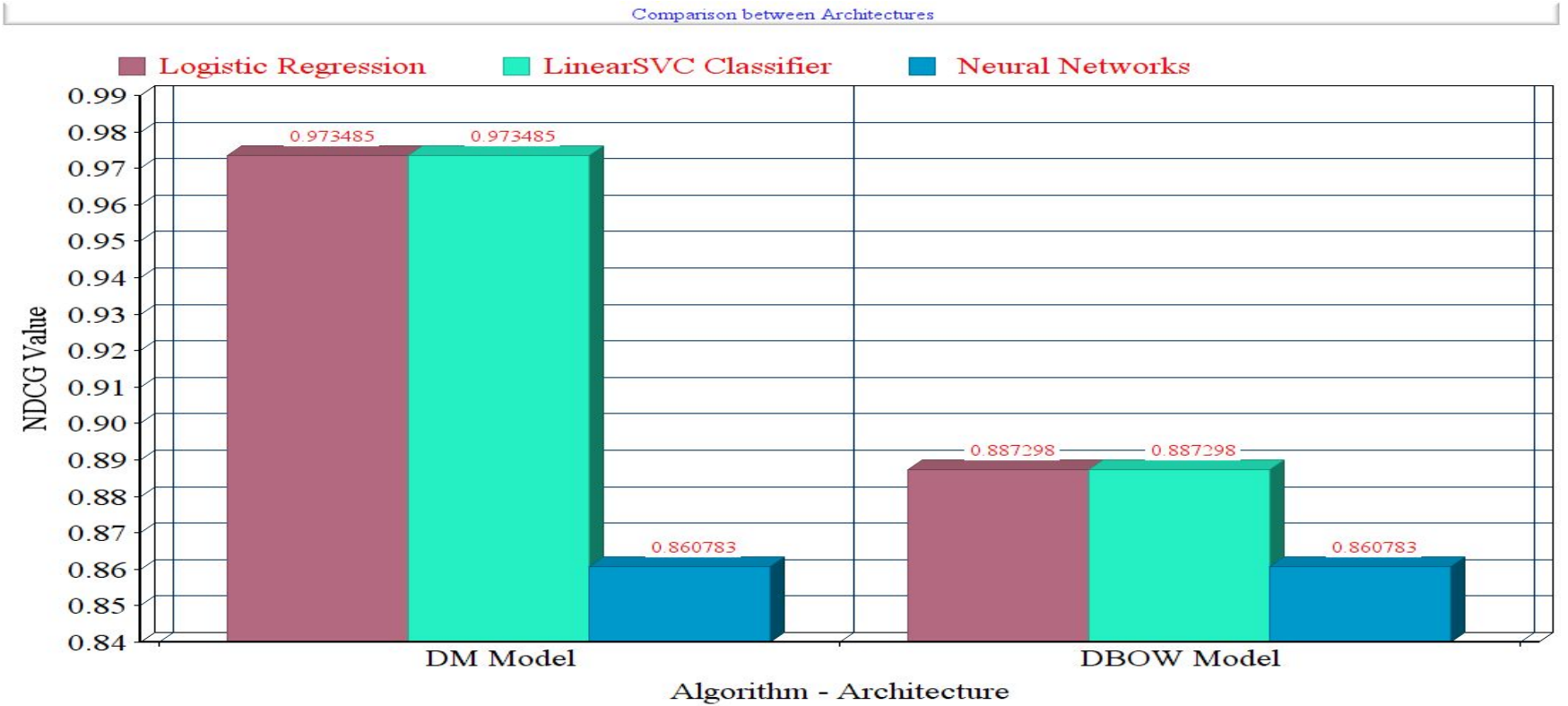
$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log(1+i)}$$

- DCG values are often normalized by comparing the DCG at each rank with the DCG value for the perfect ranking.
- It varies from 0.0 to 1.0, with 1.0 being the ideal ranking of the entities.

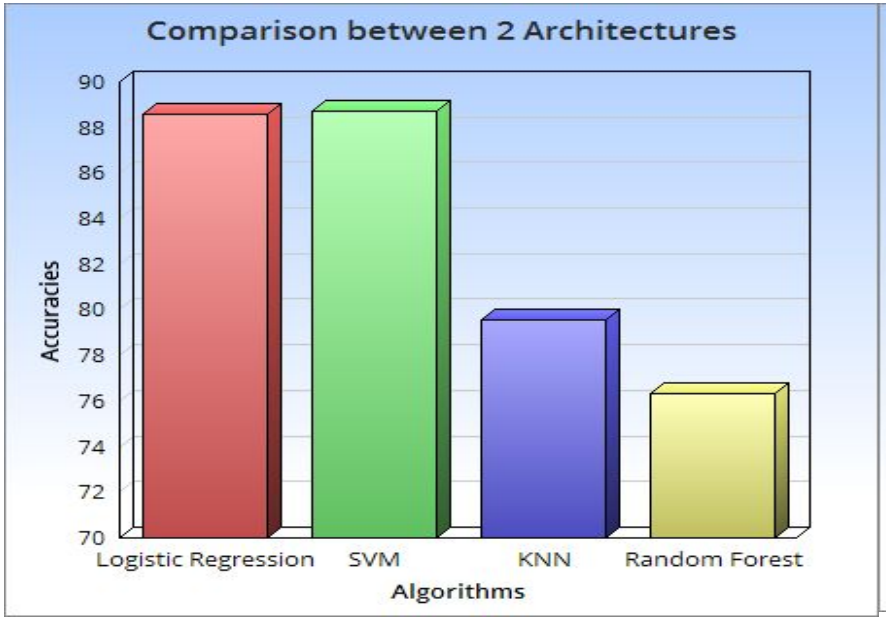
Comparison Graphs - Different Architectures (ACM)



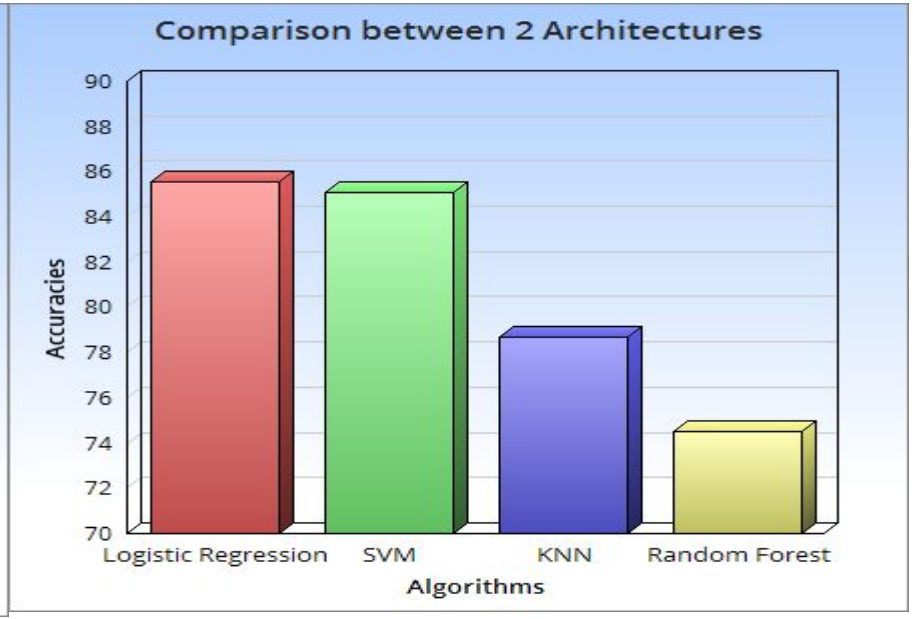
Comparison Graphs - Different Architectures(ACM)



Comparison Graphs - Different Models (IMDB Dataset)



Skip Gram



CBOW

Challenges Faced

- Multiclass and Multilabel Data, where the set of classes scales with the number of available training examples.
  - In this type of problem, the standard assumption of having a fixed set of classes is too simplistic, and straightforward generalizations of methods for binary classification (such as multi class SVM) may be impractical.
  - We used the One-vs-all classifier, where we fitting one classifier per class. For each classifier, the class is fitted against all the other classes. In addition to its computational efficiency (only n_classes classifiers are needed), one advantage of this approach is its interpretability. Since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier.

References

- https://radimrehurek.com/gensim/models/doc2vec.html
- http://rare-technologies.com/Doc2Vec-tutorial/
- https://github.com/piskvorky/gensim
- Quoc V. Le, and Tomas Mikolov, “Distributed Representations of Sentences and Documents ICML”, 2014
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR”, 2013.
- http://web.stanford.edu/class/cs276/handouts/EvaluationNew-handout-6-per.pdf

Resources Link

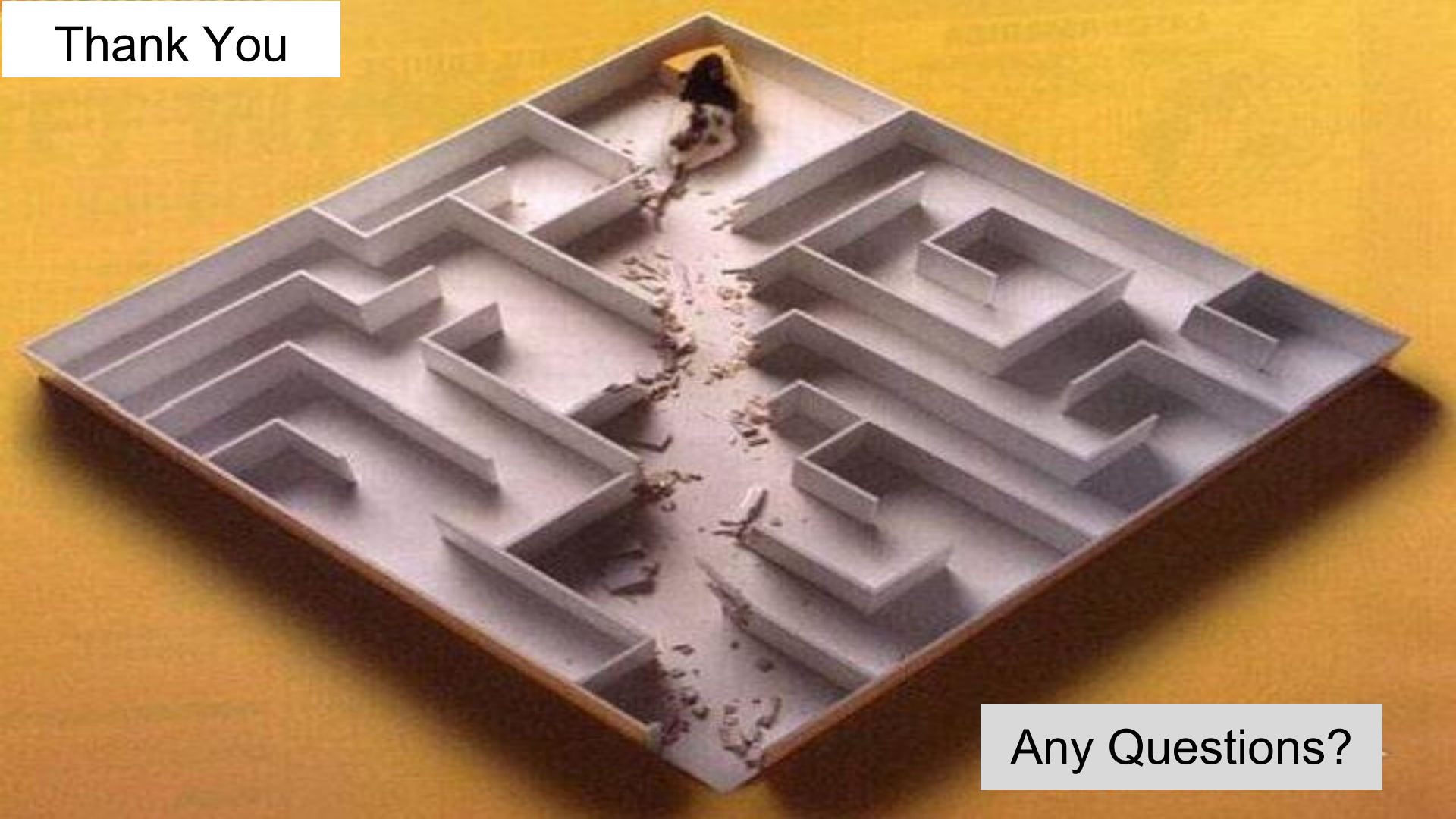
Project Webpage: http://rohitsakala.github.io/semanticAnnotationAcmCategories/

Source Code Repository: https://github.com/rohitsakala/semanticAnnotationAcmCategories

Video: https://youtu.be/706HJteh1xc

Slides: https://www.slideshare.net/secret/ELAqfEHI6F0uDq

Thank You



Any Questions?