Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 3, July 2021: 5129-5139

Research Article

Detecting Replica threads in online Q/A FORMS

Dr Suneetha Merugula[1], Dr Balajee Maram[2], Karri n v s s sai santosh reddy[3]

Abstract

The number of questions asking in online Q/A forms are rapidly increasing day by day: in which many of the questions are of same internal meaning but the formation of the question and the vocabulary used in the question are different.so recommending same answer to the near duplicate or duplicate threads are becoming really thought.In this project, the proposed system aim to develop a suitable online machine learning model to detect the question pair similarity so that the proposed system can easily detect that whether it is duplicate, near duplicate or not a duplicate question. If the question is duplicate or near duplicate of the previously existing answered question the proposed system can recommend similar answer to that duplicate thread

Keywords—duplicate threads ,machine learning model ,near- duplicate model

INTRODUCTION

This is a fascinating problem that is currently facing by many online Q/A forms. Flow gathering search advancements cannot identify strings with close copy content and amass them in the list items. Therefore, discussion clients are over-burden with copied query items and like to create new strings without attempting to discover existing ones. in which the same questions are asked by many people in which; the inner meaning of the question may be the same as many of the other questions, but the way of writing the question is different in terms of vocabulary, sentence formation and verb forms in grammar due to this problem The answer suggesting for the same multiple questions will become tougher so to avoid that the proposed system are building a machine learning model that can detect the duplicate threads, Despite their unique testing strategy, the quantity of copied models in the dataset was a lot higher than the non- copied ones, as indicated by any Q/A form sites like LinkedIn, Quora, Reddit, etc. The negative models were unsampled utilizing sets of related inquiries, inquiries regarding the subject that are not indistinguishable, to make a more adjusted dataset.to solve this problem by building a models takes few steps and few parameters as a measure of detecting duplicate threads

Data cleaning and feature Extraction

to get the best model, we need the best information that contains a one-of-a-kind arrangement of inquiry

[1] [1]Asst.Professor ,Dept. of IT, GMR Institute of Technology Rajam, Andhra Pradesh, India. suneetha.m@gmrit.edu.in
[2]Associate Professor,Dept of CSE, GMR Institute of Technology Rajam, Andhra Pradesh, India. balajee.m@gmrit.edu.in
[3]Student, Dept. of IT GMR Institute of Technology Rajam, Andhra Pradesh, India. ksr5298@gmai.com

sets. Subsequent to the cleaning of information, it is needed to extricate the needy and autonomous highlights. In the majority of the informational indexes, the primary issue is it just contains not many features like <q.no, questions,time_lapse, class>. In any case, this information won't be adequate to make the best information model. To get the best model, the proposed system need the most extreme learning features. so to get the most extreme learning features, the proposed system need to make some more sections from the all-around existing information .like consider the above situation that the proposed system have just two inquiries and no other data is available. the fundamental way to deal with getting more data is we can part the inquiries dependent on some parameters, and we can get some other new features like length, root words, most incessant words, and so forth

Business constraints and objectives

Before building any model, the proposed system need to consider all the business constraints and objectives. As it is a duplicate pair detection, the output should be very accurate. Let us consider two questions, Q1 and Q2, and consider Q1 and Q2 are two different questions, but our machine learning model predicts both are the same. But end- users are ingenious enough to understand the language. So if the user gets to know that both are different but it is recommending to the question. then it will affect the company’s reputation. so a well-defined confidence interval should be present like
P(Q1,Q2,similarity)>0.99. if this condition satisfies then the proposed system can consider the model

Performance metrics

The primary key performance indicator for any of the duplicate question pair similarity is log loss, as it internally contains probability scores as a measure and the second performance indicator is a binary confusion matrix which tells us about the various errors that are happening. the lesser the log loss value, the better the model in a binary class classification if the class is 1 and probability of the class is close to 1 and the lesser the value. The better the model. Similarly for the negative class

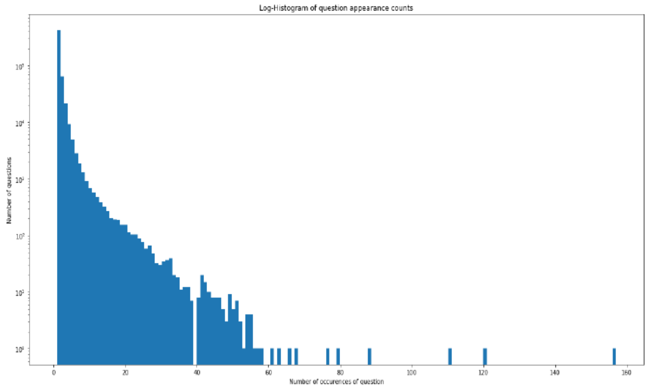
$$H_p = -1/n \sum (yi * log(pi) + (1 - yi) * log(1 - pi)$$

PROPOSED SYSTEM

EXPLORATION

The proposed system is considering a publicly available Kaggle data set to train our model because collecting millions of question pairs will be a lifetime task for anyone. This dataset, before prepossessing, contains only six columns, but this information is not at all sufficient to derive the data needed, so the proposed system need to add some more features derived from the existing elements called derived attributes.
the features are
Train.csv contains 5 columns : qid1, qid2, question1, question2, is duplicate
Size of Train data - 60MB
No of rows in Train data = 404,290
The character set of our dataset was not carefully ASCII; the proposed system tracked down that

6,228 inquiries contained non-ASCII characters and these inquiries happened across 8,744 inquiry sets. There were additionally, two sets that contained a empty string for one of their inquiries.Below is the Log-Histogram of question appearance counts. Which tells us Maximum number of times a single question got repeated



.

TRAIN AND TEST CONSTRAINT

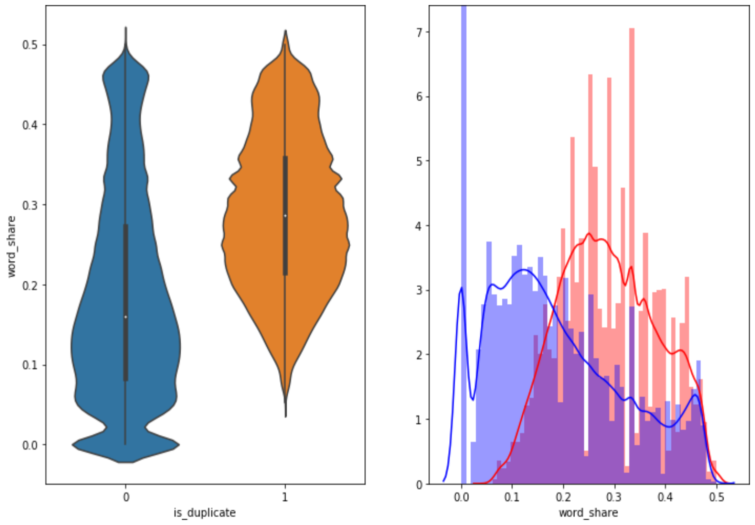
1.BLIND-SPLIT

If there is a timestamp in the given dataset, then time-based splitting will be preferred because things change along with time. if there is no timestamp in the dataset, then normal 70:30 or 80:20 splitting is better to use. Without proper data cross-validation, there will be a problem of data leakage

FEATURE EXTRACTION

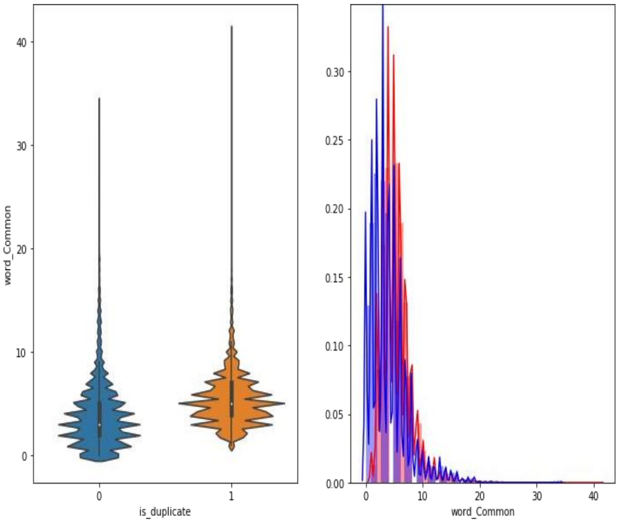
As the proposed system are dealing with the duplicate question pair threads many of the Datasets don't have lot of Features that are required for the analysis.as the proposed system know that more the features better the model will be, so in order to make the model better, the proposed work will add few other parameters to it like

1.frequency of quention_id1
2.frequency of quention_id2
3.Length of question1
4.Length of question2
5.No of words in the Question1
6.No of words in the Question2
7.Common Words in both Question1 and Question2



The above figure depicts the relation between duplicate questions and word share. The distributions for normalized word_share have some overlap on the far right-hand side, i.e., there are quite many questions with high word similarity.

Moreover, the Avg word Share and common words are more they are counterfeit.



The circulations of the word_Common highlight in comparable and non-comparable inquiries are profoundly covering.

Advanced feature extraction (NLP and Fuzzy Features):

As the proposed system is dealing with the text data and have a lot of unnecessary text which do not give us actual point to point information, so there are some techniques we are using to for better features and comparison of words

Tokens are the words that are derived from splitting the sentence considering space as a delimiter. Stop Words need to be deleted based on NLTK, and we can add/remove from the standard list based on NLTK

which are not stop words are considering as a word Some features are::

Com word count to min length of word count of Q1 and Q2

Com word count to max length of word count of Q1 and Q2

Com stop word count to min length of stop count of Q1 and Q2

Com stop word count to max length of stop count of Q1 and Q2

All these above features are used to extract the maximum data from the 5 column dataset

Fuzzy wuzzy

Fuzzy wuzzy is a python based library in which internally it uses fuzzy logic mechanisms .To tell the similarity between the words. But there are still lots of disadvantages in it.

S1:vizag mets S2: vizag mates

In fuzzy wuzzy system it gives the similarity as 96%, but actually there are very different words in the internal meaning

TF-IDF

$$0 < TF(Wi, Rj) < 1$$

where Rj is a records and Wi are words and each record is binary bag of words(vec)
term frequency always lies between 0 and 1 and this tfidf techniqies used in informational retrieval domain

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}},$$

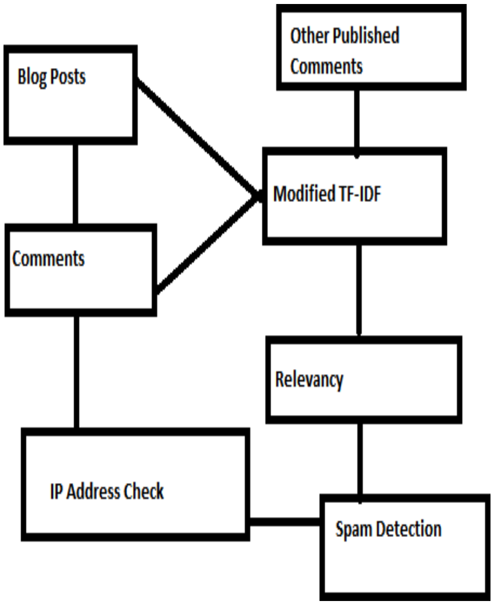
And IDF is only for word in the whole data corpus.to be more precise it tells how often a word occurs in the whole corpus.

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

If Wi is more frequent in the corpus then IDF decreses and vice versa also possible and same relation works for Ni and IDF

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

If we multiply both TF and IDF the TF says how often word occurs in record and IDF says How rarely the word occurs in the corpus as the rare words have the most preference.at the end the proposed system are giving more preference to the least occuring words in data corpus
This method works fine but it doesn’t take the semantic meaning



Word2vec for semantic meaning detection

This technique converts text to the vector form and unlike tf-idf and bow it also tells us the semantic meaning of the vectors. The word2vec initially converts any word as a dense vector if 2 words are similar then 2 vectors are also similar and vice versa also possible.it is capable of generating relations also. This word2vec learns relationships automatically from raw text it works better if our data corpus is very big generally newspaper data.intutively word2vec works on the technique of neighborhood. That is if we want to compute the neighborhood of word vector Wi it will consider all the neighborhood vectors

$$W = [w1, w2, w3, ...Wi, ...wn]$$

$$W2V(w_i) = argmax(neighborsimilarity(W))$$

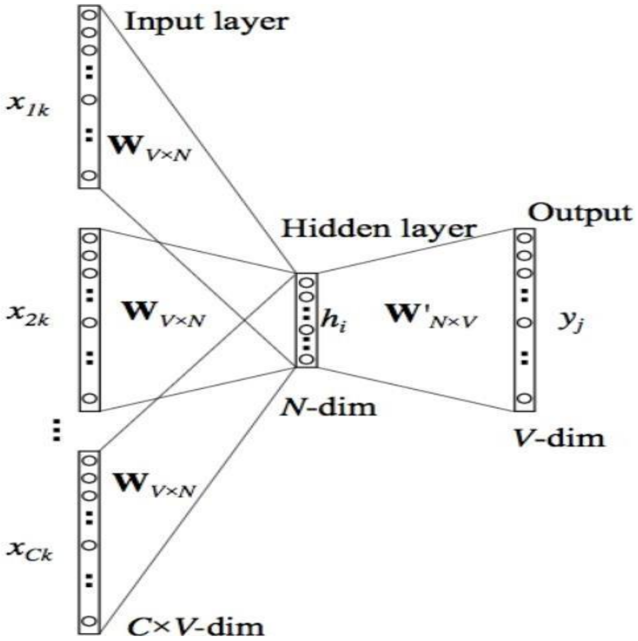
If N(Wi)=N(Wj) then Vi=Vj

Word2Vec :CBOW APPROACH

Let us consider a Word vector

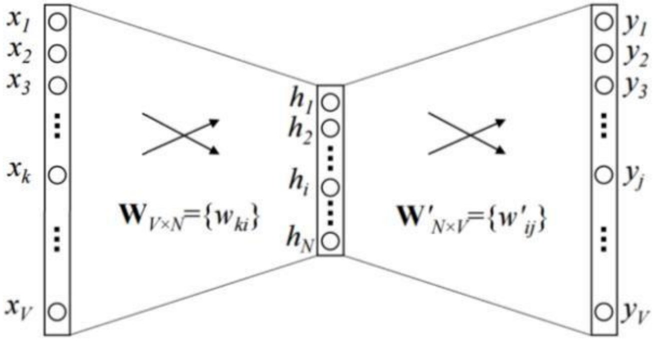
$$W = [w1, w2, w3, ...Wi, ...wn]$$

In which our focus word is Wi and remaining all are context words. The core idea behind it is the context words helps to understand the focus words better and vice versa is also possible.



As the above diagram depicts each word is v dimensional one hot encoded vector and each vector is a context word Ci.all these C*V dimensional input is connected to a N- dimensional hidden layer which uses linear activation function as we need to find the focus word based on the context words so the output vector is a v-dimensional binary vector and the activation unit that is used is softmax. After using softmax classifier it became a simple probability based regression problem.

We can also predict based on matrix factorization method instead of using this neuron method.

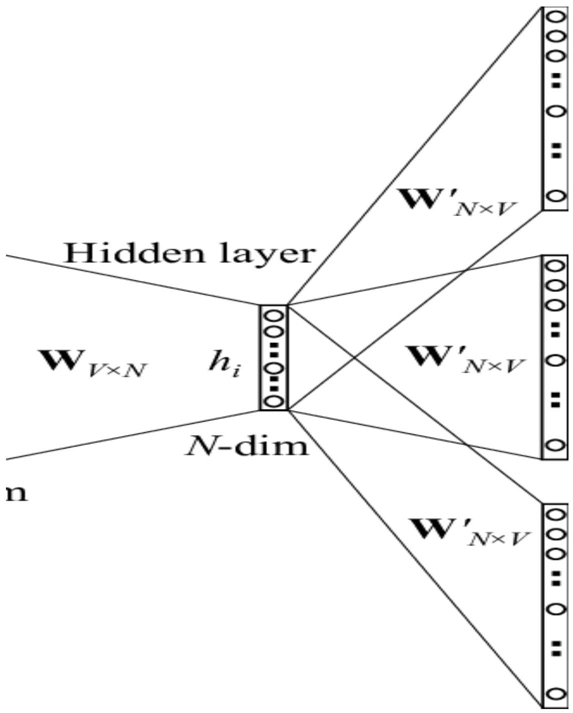

x_1
x_2
x_3
x_k
x_V
h_1
h_2
h_i
h_N
y_1
y_2
y_3
y_j
y_V
$\mathbf{W}_{V\times N}=\{w_{ki}\}$
$\mathbf{W}'_{N\times V}=\{w'_{ij}\}$

Word2Vec :SKIP-GRAM APPROACH

Let us consider a Word vector

$W = [w1, w2, w3, ...Wi, ...wn]$

In this method the proposed system will predict context words based on the focus word which is opposite to the CBOW


$\mathbf{W}'_{N\times V}$
Hidden layer
$\mathbf{W}_{V\times N}$
h_i
$\mathbf{W}'_{N\times V}$
N-dim
n
$\mathbf{W}'_{N\times V}$

We have only 1 input vector(one hot encoded) and it will be given to N-dimensional hidden layer with some linear activation now based on focus word the proposed system need to find context words i.e

Ci=[C1,C2,C3,….Ci,…..Cn]

the proposed system will connect hidden layer to the soft max that is trying to predict Ci so in the output layer it have K softmax.it is simply like a K-Multi class Classifier

In both CBOW and Skipgram the proposed system have same no of weights i.e (K+1)(N*V)

But it differs in no of softmax functions as we need K softmax functions in the Skipgram it is computationally hard so we are going with CBOW

If our data is small then skipgram is advisable than CBOW

GLOVE SEMANTIC MEANING DETECTION

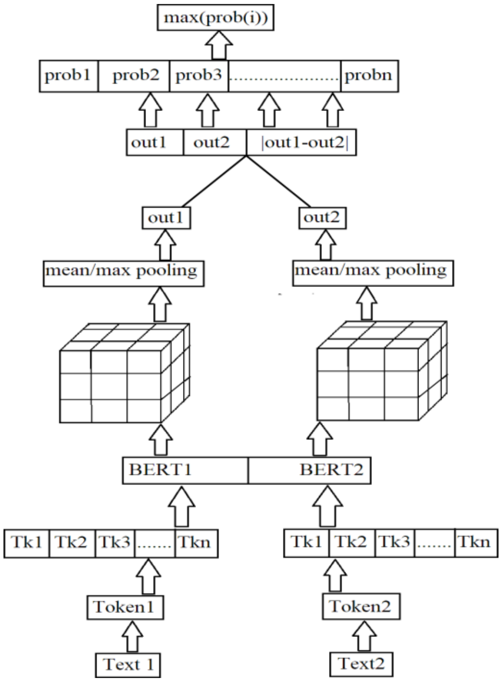
It is one of the summation of count based models and predictive modelling approach.in this model the proposed system are taking advantage of whole global statistics unlike word2vec. the proposed work can derive the relationships between words from the matrix The version rests on a instead easy concept that ratios of phrase-phrase co- occurrence possibilities have the capacity for encoding some of which means which may be encoded as vector differences. Therefore, the learning goal is to analyze phrase vectors such that their dot product equals the logarithm of the words' possibility of co-occurrence matrix. The model makes use of the principle benefit of remember data — the capacity to seize international statistics — even as concurrently capturing the significant linear substructures common in current log-bilinear prediction-primarily based totally strategies like word2vec. As a result, GloVe will become a international log-bilinear regression version for the unsupervised gaining knowledge of word representations that outperforms different models on word analogy, word similarity, and named entity reputation tasks.

$$P_{ij} = P(j|i) = \frac{X_{ij}}{\sum_{k \in context} X_{ik}}$$

SENTENCE SIMILARITY USING SIAMESE BERT NETWORK

From 2010 to 2020 has been an affectation point for AI models that deal with text based data .Bert is an advanced model of regular Transformer(ULM-FIT) and openAI(ELMo) the proposed system can train the BERT in two steps BERT with Semi supervised training on huge amounts of textual data like books,Wikipedia articles etc and another one is Supervised training on a specific task labeled dataset.bert contains a set of encoders.unlike rnn it is capable of taking multiple input vectors at a time.

Our Text data is will be first tokenize into tokens for this we need to perform tokenization and after the tokenization we need to add [CLS] and [SEP] tokens.and the next step of tokenization is substituting tokens with their ids and then the input of id's will be given to Siamese BERT model and the output is in the matrix format that contains 768 hidden units if we want to be classified we can use classification algorithms or softmax based on the probability scores if the proposed system want any regressor score then simple cosine similarity is enough



The ouput of the BERT is a matrix with 768 hidden units as each row is associated with dataset the proposed system are interested in converting 3D matrix to a 2D matrix for that we are applying max pooling or mean pooling and the outputs of the corresponding max pooling vectors may be closer to each other or father from each other for that the proposed system need to check with cosine similarity. If the cosine similarity is very close value then those two sentences are related to each other else they are farther from each other

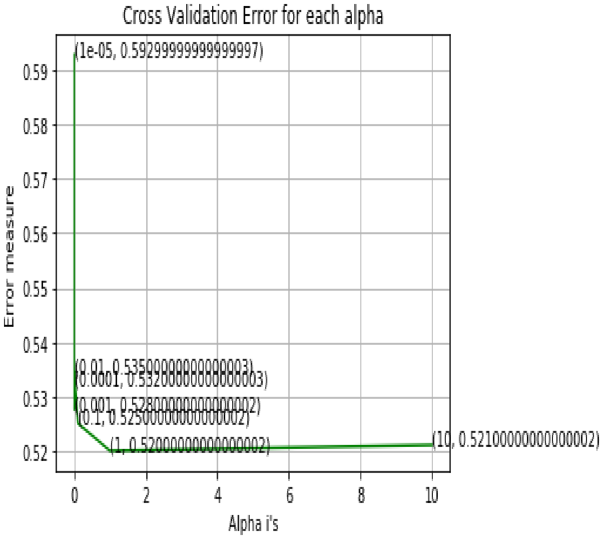
$$o = \mathrm{softmax}(W_t(u, v, |u - v|))$$

If you want to classify wheater they are similar or not then use softmax classifier and take the max probability embeded word.

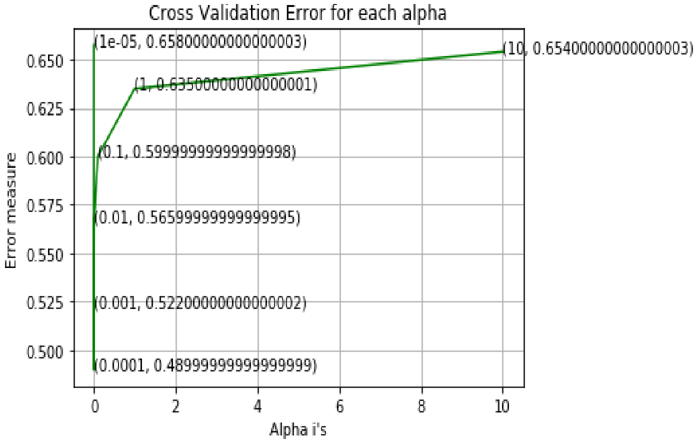
EXPERIMENTAL RESULTS

Model Class	**Test Results**		
	Model	**Accuracy**	**F-Score**
Linear	1.Logistic regression with unigrams 2. Logistic regression with bigrams 3.SVM with unigrams 4.SVM with bigrams	78.90 79.52 79.94 81.00	63.81 72.60 73.00 70.21
Tree based	Decision Trees Random forest	74.38 76.24	66.24 68.35
Ensemble models	XG-boost Adaboost	81.63 83.65	77.8 75.6

Logistic Regression cross validation error for set of 30000 points



The train log loss is: 0.513842874233 The test log loss is: 0.520035530431 Total number of data points : 30000
Logistic SVM cross validation error for set of 30000 points



For best alpha = 0.0001 The train loss is: 0.47805467728 For best alpha = 0.0001 The test loss is: 0.489669093534 Total data points : 30000

REFERENCES

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of EMNLP

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations (ICLR).

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. arXiv preprint arXiv:1904.09675.

] Jiang Zhao, Tian Tian Zhu, and Man Lan. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In SemEval, 2014

BLOG-POSTS

THE ILLUSTRATED BERT, ELMO, AND CO. (HOW NLP CRACKED TRANSFER LEARNING) BY JAM ALAMMER

A VISUAL GUIDE TO USING BERT FOR THE FIRST TIME BY JAM ALAMMER

THE ILLUSTRATED WORD2VEC BY JAY ALAMMER

THE ILLUSTRATED TRANSFORMER BY JAY ALAMME