Detecting Duplicate Question Pairs with Quora

James Peng
U.C. Berkeley
School of Information
james.peng@ischool.berkeley.edu

Brad Putman
U.C. Berkeley
School of Information
bwputman@ischool.berkeley.edu

Geoffrey Link
U.C. Berkeley
School of Information
geoffrey.link@ischool.berkeley.edu

Abstract

An interesting research question in NLP is how to classify whether or not two natural langage questions are duplicates of each other. At present, Quora uses a random forest model to accomplish this task. Having an accurate duplicate question detection model makes it easier to find high quality answers to questions; the result of which is an improved experience for Quora writers, seekers, and readers. In this paper we discuss the factors that make identifying duplicate questions difficult, we highlight previous research to provide a foundational understanding and justification for how we approach the problem, and we implement a few potential solutions and discuss their results. We have also provided access to our GitHub repository to supplement this paper.

1 Introduction

During the last decade internet based Collaborative Question Answering (CQA) platforms have increased in popularity. These platforms offer a social environment for people to seek answers to questions, and where the answers are offered by other community members. Users pose questions in natural language - as opposed to queries in web search engines - and community members propose answers in addition to voting and rating the information posted on the platform.

This content has attracted the attention of researchers from a number of domains who aim to automatically return existing, relevant information from the CQA database when a novel question is submitted. Proposed approaches fall into two categories:

1. Determining the most helpful answers to a given question

2. Determining similar questions

Over 100 million people visit Quora every month, so it is no surprise that many people ask similarly worded questions. Multiple questions with the same intent can cause seekers to spend more time finding the best answer to their question, and make writers feel they need to answer multiple versions of the same question. Quora values canonical questions because they provide a better experience to active seekers and writers, and they offer more value to both of these groups over the long term.

2 Background

The main difficulty in detecting duplicate questions is the lexical chasm - a phenomenon where semantically similar questions are lexically dissimilar. An example of this occurrence is the interchangeable pair of questions “Where can I watch movies online?” and “Are there any websites for streaming films?” The opposite case is also possible; questions having words in common may have different semantic meanings. Besides the need for accurately identifying a question’s semantics, a solution to the problem must deal with noisy information such as misspelled words, polysemy, and short ungrammatical questions.

Duplicate questions are typically found by comparing the query questions to the context of existing questions as it has been shown that finding duplicate questions based solely on their answers does not perform well. However, combining information derived from existing questions and their answers outperforms many other strategies. In recent years, topic modeling has been applied to this problem as it reduces the dimensionality of textual information when compared to classical methods

(such as bag-of-words) and efficiently handles polysemy and synonymy. [Zhang2014]

3 Related Work

Detecting semantically equivalent sentences or questions has been a longstanding problem in natural language processing and understanding. The principal challenge when retrieving related questions and answers in a CQA database given a new question is the lexical gap that may exist between two semantically similar questions. In general, a method that intends to solve the problem of question retrieval should be composed of at least two main parts: a document representation that can properly express the semantics and context of QAs in the database; and a mechanism for comparing the similarity of documents given their representations. The most widespread document representation methods in the literature are those based on the bag of words (BoW) concept, which explicitly represents each of the document's words. Comparison is achieved by computing the number of matching words between two BoW representations.

There exist several variations of the BoW class of methods, each weighting words that have specific properties in the dataset. This class of methods is able to measure two documents' lexical similarity but does not capture information regarding semantics or context. BoW obtains the similarity between two questions by computing the number of the similar words in them. Despite their successes, BoW methods only capture the string matching features in computing text similarity and overlook word sense, word order, and semantic information.

In CQA databases, questions and answers are often short and contain many word variations resulting from grammatical inflection, misspelling, and informal abbreviations. As a consequence, BoW representations in CQA corpora produce a vector representation that can be too sparse. Besides sparsity, BoW representations do not provide a measure of co-occurrence or shared contextual information, which can increase the similarity of related documents.

Convolutional neural networks (CNN's) utilize layers with convolving filters that are applied to local features. Originally invented for computer vision, CNN models have subsequently been shown to be effective for NLP and have achieved excellent results in semantic parsing, search query retrieval, sentence modeling, and other traditional NLP tasks [Kim2014].

As Santos et al. [Santos2015] observed: 1) BoW combined with a Convolutional Neural Network (BoW-CNN) is more effective than BoW based information retrieval methods such as TF-IDF; 2) BoW-CNN is more robust than pure CNN for long texts. Santos proposes a hybrid neural network architecture that combines a traditional BoW representation with a distributed vector representation created by a CNN to retrieve semantically equivalent questions. Santos' BoW-CNN is evaluated over two different CQA communities in the Stack Exchange site, and compared against CNN and six well-established information retrieval algorithms based on BoW. Santos' results show that BoW-CNN outperforms BoW-based information retrieval methods such as TF-IDF in all evaluated scenarios. Further, Santos is able to show that for short texts (title of the questions), an approach using only CNN obtains the best results, whereas for long texts (title and body of the questions), the hybrid approach (BoW-CNN) is more effective.

Interestingly, Kim [Kim2014] through a series of experiments with CNN's trained on top of pre-trained word vectors for sentence-level classification tasks found that a simple CNN with little hyperparameter tuning and static vectors achieves excellent results on multiple benchmarks. Learning task-specific vectors through fine-tuning offers further gains in performance. The Kim's CNN models improved upon the state of the art in 4 out of 7 tasks, which include sentiment analysis and question classification.

Traditionally, within natural language processing, much of the work with deep learning methods has involved learning word vector representations through neural language models and performing composition over the learned word vectors for classification. As Kim identifies, Word vectors, wherein words are projected from a sparse, 1-of-V encoding (where V is the vocabulary size) onto a lower dimensional vector space via a hidden layer, are essentially feature extractors that encode semantic features of words in their dimensions. In such dense representations, semantically close words are likewise close in euclidean or cosine distance in the lower dimensional vector spaces.

In the last decade, as Zhang et al. [Zhang2014]

illustrate, topic modeling has become an important method for text analysis. Since the topics that characterize a document can be considered a semantic representation, it is possible to use topic distributions inferred using a method such as Latent Dirichlet Allocation (LDA) to measure the semantic similarity between documents in a corpus. Unfortunately, the Quora dataset does not lend itself well to topic modeling as the questions are short and unstructured. Thus, dimensionality reduction by topic would prove difficult.

As Dey et al. [Dey2016] demonstrate, traditional machine learning algorithms such as Support Vector Machines (SVM's) using hand-picked features and extensively preprocessed data perform well on the SemEval-2015 dataset. They argue that the performance of deep learning methods is heavily limited by the small, noisy datasets that they are trained on. Dey proposes a feature-set driven approach with SVM-based machine learning that relies upon upon lexical, syntactic, semantic, and pragmatic features of a given pair of tweets, to label whether two tweets are paraphrases of one another.

Deep learning techniques have made considerable progress in recent years. As Yuan et al. [Yuan2016] have identified, determining the intended sense of words in text - word sense disambiguation (WSD) - is a longstanding problem in natural language processing. Recently, researchers have shown promising results using word vectors extracted from a neural network language model as features in WSD algorithms. However, a simple average or concatenation of word vectors for each word in a text loses the sequential and syntactic information of the text. Thus, in the aforementioned paper, Yuan studies WSD with a Long Short-term Memory (LSTM) neural network to better capture the sequential and syntactic patterns of the text and demonstrates state-of-the-art results, especially on verbs.

4 Dataset

We worked with the "Quora Question Pairs" data set from the corresponding Kaggle Competition. The data set contains 404,302 pairs of actual Quora questions, and 149,264 (approximately 37%) are marked as duplicates. It is important to observe that the duplicate tags are subjective and therefore vulnerable to some level of bias and error. The questions themselves are uncleaned (containing typos and nonsense words) and have lengths ranging from a single character to 271 words. There are 2,345,806 pairs of Quora questions in the test data set.

The ground truth is the set of labels that have been supplied by human experts. The ground truth labels are inherently subjective, as the true meaning of sentences can never be known with certainty. Human labeling is also a 'noisy' process, and reasonable people will disagree. As a result, the ground truth labels on this dataset should be taken to be "informed" but not 100% accurate, and may include incorrect labeling. We do believe that the labels - on the whole - represent a reasonable consensus.

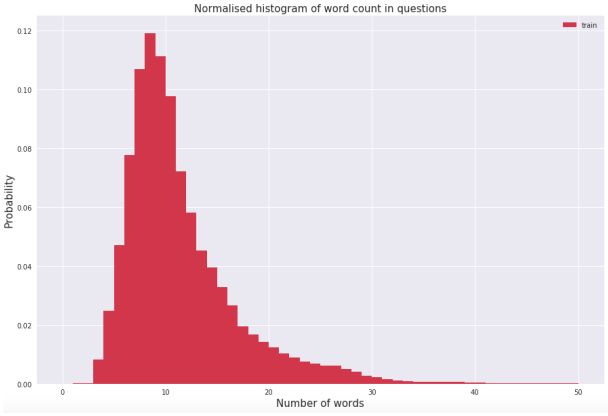

Figure 1: Normalized histogram of word counts.

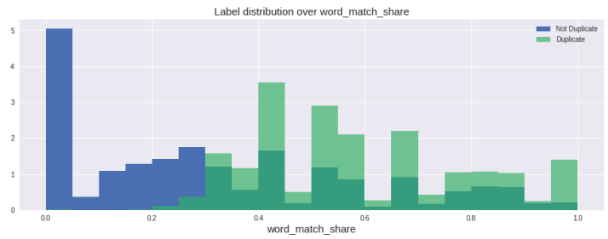

Figure 2: Words shared between Duplicate and Non-Duplicate questions.

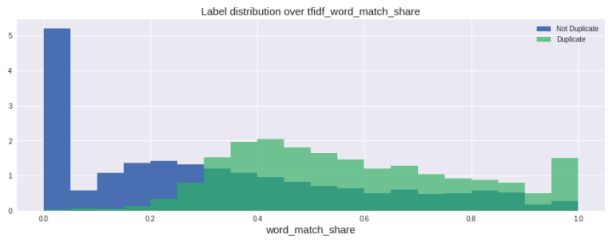

Figure 3: TF-IDF words shared between Duplicate and Non-Duplicate questions.

5 Modeling

5.1 Model Selection

Since neural techniques do well at capturing semantic meaning to avoid polysemy-induced false positives, we decided to use two different neural network architectures as the basis of our efforts. We were confident that the Quora dataset was large enough to support a deep learning approach.

Given Santos' and Kim's results, we chose to use a CNN as our first model combined with vectors from Stanford's Global Vectors for Word Representation. Due to the brevity of the questions in the Quora dataset, these papers suggested that pure CNN model would perform better than a BoW-CNN hybrid.

For our second model, inspired by Yuan's research, we used a LSTM neural network to attempt to consider words in context without actually having to model that context. The LSTM also used pre-trained Word2Vec embeddings to determine similar meaning between words

To balance out our bias towards neural networks, we also implement an XGBoost model as our third classifier. The XGBoost model was employed with a number of distance metrics and Subject-Verb-Object feature information, and served as a "control" model to determine if a linear model could be competitive versus neural networks. We were curious if these human-engineered metrics alone could provide enough signal in the data to yield accurate classifications.

5.2 Word Embeddings

As previously stated, our LSTM model leveraged Word2Vec embeddings to guage word meanings. We used Gensim combined with Google News' pre-trained Word2Vec corpus executed against both the training and test corpora for our LSTM model.

Our CNN model used Stanford's Global Vectors for Word Representation (GloVe). Our goal here was to distinguish between questions wherein a single word changes the entire context of the question. For example, "How do bartenders become bartenders in Texas?" must be considered a different question than "How do bartenders become bartenders in California?".

5.3 Subject-Verb-Object Triples

Subject-Verb-Object (SVO) is an important property of sentence structure because it encapsulates the semantic meaning of a sentence. In one Quora question, there could be one or more SVO triples. For example, "he and his brother shot me and my sister" has 4 SVO triples: "he shot me", "he shot sister", "brother shot me", and "brother shot sister".

In this project, we used an open-source python module to extract SVO triples. Once the SVO triples were extracted, we engineered features such as the count of SVO triples, as well as the percent of common subject, verb, and object.

It is worth noting that SVO triples can be also feature-engineered through a Vector-Space-Model (VSM) like Word2Vec. Once SVO's are represented as vectors, many distance-based metrics can be calculated as new features. Unfortunately, we did not have time to implement this approach.

6 Results and Discussion

In analyzing model performance, we placed higher emphasis on comparing and contrasting how each model learned to understand natural language than we placed on tuning each model to achieve an optimal score on the test data set. The table below illustrates the amount of classification overlap in each of the models we evaluated.

Model Comparison	Classification Similarity
CNN vs. LSTM	92.97 %
LSTM vs. XGB	74.44 %
CNN vs. XGB	72.12 %
CNN vs. LSTM vs. XGB	69.76 %

By digging into the question pairs that were evaluated differently by different models, we gained some insight into the classification processes themselves. We randomly sampled from those question pairs with labeling discrepencies and noticed that the three models exist on a spectrum ranging from high semantic emphasis to high syntactic emphasis. The LSTM placed the highest emphasis on semantic meaning - it showed great aptitude in general topic matching, but was vulnerable to overgeneralizing, and thus it showed a higher rate of false positives. On the other end of the spectrum, the XGBoost model placed the high-

est emphasis on syntactic representation - it was the best at particularization, but often fell victim to the lexical chasm and a high rate of false negatives. The CNN fell in between the other two models on the semantics/syntax spectrum. An important caveat with this interpretation is that we are merely judging the order of these models on the described spectrum; we do not make any claims about relative "distance" between them.

Our final scores on the test set (expressed as negative log loss) were 0.41 for the XGBoost model, 0.35 for the CNN model, and 0.33 for the LSTM model. The XGBoost model had a significantly faster runtime than the CNN and LSTM models. We believe the neural networks are less prone to bias because their features were not manually engineered. Another important factor to note is that the LSTM's sequential learning methodology allows the model to get better over time by adding additional training epochs after the inital training phase; this fact makes it a lower-maintenace model than the XGBoost and the (feed-forward) CNN because they cannot incorporate new training data without being retrained from scratch. One final factor to note is our belief that false positives are preferable to false negatives in the context of duplicate question detection; consequently, the usefullness of incorrectly labeled question pairs leans in favor of the nerual nets.

In conclusion, the results gathered from our three models support those mentioned in earlier papers on this topic. After taking into account all of the factors listed in the previous section, we have concluded that the LSTM neural network is the best machine learning model for duplicate question detection.

References

[Santos2015] Ccero dos Santos, Luciano Barbosa, Dasha Bogdanova, Bianca Zadrozny. 2015. *Learning Hybrid Representations to Retrieve Semantically Equivalent Questions*. School of Computing, Dublin City University, Dublin, Ireland. http://www.aclweb.org/anthology/P15-2114.

[Dey2016] Kuntal Dey, Ritvik Shrivastava, Saroj Kaushik. 2016. *A Paraphrase and Semantic Similarity Detection System for User Generated Short-Text Content on Microblogs*. IBM Research India. https://pdfs.semanticscholar.org/f4f0/721d0eea03ed531f603d98cf7bb3b9ed72b9.pdf?_ga=2.168648929.1302613579.1501005165-460806673.1497050040.

[Yuan2016] Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, Eric Altendorf 2017. *Semi-supervised Word Sense Disambiguation with Neural Models* Google. https://arxiv.org/pdf/1603.07012.pdf.

[Kim2014] Yoon Kim 2014. *Convolutional Neural Networks for Sentence Classification* New York University. https://arxiv.org/pdf/1408.5882.pdf.

[Zhang2014] Wei-Nan Zhang, Ting Liu, Yang Yang, Liujuan Cao, Yu Zhang, Rongrong Ji 2014. *A Topic Clustering Approach to Finding Similar Questions from Large Question and Answer Archives* New York University. http://journals.plos.org/plosone/articleid=10.1371/journal.pone.0071511#s1.