Determining Duplicity of pairs of Questions: Integrating Statistics and Sentiment Analysis

Saurabh Singh
Northeastern University
singh.saurab@husky.neu.edu

Abstract

The paper proposes an integrated method of statistics and sentiment analysis to determine the duplicity of pairs of questions. The key to the method is to generate the training data comprising of 3 features – a) cosine similarity per pair of questions b) a sentiment value per question. Logistic Regression model based on these three features is used to predict whether two questions are duplicate or not. In the preliminary experiments, the naïve method of using the cosine vector similarity to predict the duplicity gave 67% accuracy. Hence it is integrated with the sentiment value of the questions. Through a series of experiments, it was found that the accuracy jumped by 4%.

1 Introduction

Matching two texts is central to many natural language applications, such as machine translation (Brown et al. 1993), and question and answering (Xue, Joen and Croft 2008).

Quora and Stack Overflow are the question-and-answer site where questions are asked, answered, edited and organized by its community of users. An important product principle for them is that there should be a single question page for each logically distinct question. For instance, the queries “What is the most populous state in the USA?” and “Which state in the United states has the most people?” should not exist separately on Quora because the intent behind both is identical. Having canonical page for each logically distinct query makes knowledge-sharing more efficient in many way: knowledge seekers can access all the answers to a question in a single location, and writers can reach a larger readership than if that audience is divided amongst several pages.

The problem of identifying similar questions is however different than simply matching two texts. Consider these two questions: “Is being bored good for a kid?” and “Is being a good kid and not being a rebel worth it in the long run?”. A state of the art algorithm (Microsoft text analytics API) suggests that they are similar with 68% confidence, which as we can see is not actually the case. What are we missing here? I believe the thought and feeling that goes along with the question have an impact too, i.e the sentiment associated with the asked question.

Inspired by this belief, the paper proposes to view the problem of question matching as an integrated statistics and sentiment analysis problem and solve the same. The statistical method of TF-IDF (Salton, Fox, and Wu 1983) is a widely-used method in text mining. In this method, each text is represented as a |V|-dimensional vector with each element stands for the TF-IDF score of the corresponding word in the text, where |V| is the vocabulary size. In this paper IDF (inverse document frequency) score is calculated in the whole dataset. The final matching score feature is produced by the inner product of the two vectors (Sidorov, Gelbukh, Adorno, Pinto 2014). Specifically, given two texts $T_1 = (w_1, w_2, . . . , w_m)$ and $T_2 = (v_1, v_2, . . . , v_n)$, the degree of matching is measured as a score produced by a scoring function on the representation of each text:

$$match(T_1\,, T_2) = F\big(\phi(T_1), \phi(T_2)\big)$$

where w_i and v_j denotes the i-th and j-th word in T_1 and T_2 respectively. ϕ is a function to map each text to a vector, and F is the scoring function for modelling the interactions between them.

Another two features pertaining to sentiment value per question is created using neural network based Microsoft’s sentiment analysis API (Chew-Yean Yam 2015). Since sentiment analysis needs large data for training, I directly used their models on the test data. Logistic Regression model in R is then applied to these 3 features to determine the duplicity of the question.

2 Related Work

Most previous work on text matching tries to find good representations for a single text, and usually a simple scoring function to obtain the matching results. Examples include Partial Least Square (Wu, Li and Xu 2013), Canonical Correlation Analysis (Hardoon and Shawe-Talor 2003) and some deep models such as DSSM (Huang et al. 2013), and CDSSM (Gao et al. 2014; Shen et al. 2014).

Recently a brand-new approach focusing on modeling the interaction between two sentences has been proposed and gained much attention, examples include Deep Match (Lu and Li 2013) and Syntax-Aware Multi-Sense word embedding for Deep compositional model of meaning (Cheng and Kartsaklis 2015). The proposed model in this paper fall into this category, thus it becomes important to discuss the differences between them.

2.1 Deep Match

It uses topic model to construct the interaction between two texts, and then make different levels of abstractions by a hierarchical architecture based on the relationships between topics. Compared with the integrated model of matchings matrix defined at word level and sentiment analysis, Deep Match uses topic information for determining similarity.

2.2 Multi-Sense word embedding

It uses a compositional distributional framework based on a rich form of word embedding that aims at facilitating the interactions between words in the context of a sentence. Embedding and composition layers are jointly learned against a generic objective that enhances the vectors with syntactic information from the surrounding context. Furthermore, each word is associated with several senses, the most plausible of which is selected dynamically during the composition process.

3 Project description

The dataset consisting of 400,000 lines of potential questions duplicate pairs is downloaded from Quora's website. Below is the sample data from the file:

question1	question2	dupl
Should I buy a car?	What does manipulation mean?	0
What can make Physics easy to learn?	Is physics difficult?	0
How can I be a good geologist?	What should I do to be a great geologist?	1

Data cleaning is performed. All records with any NAN values are removed. Text processing followed: a) text is converted to lower case to maintain the uniformity b) stop words are removed from the text c) stemming of the corpus is done to convert words to their root form.

A lookup table of all the unique words in the questions set is maintained. Value associated with the words are the product of term frequency and the inverse document frequency.

$$Term\ Frequency \quad t.f = \frac{f_{t,d}}{\sum_{t'\,\varepsilon\,d} f_{d,t'}}$$

where $f_{t,d}$ is the raw frequency count of the text in all the document. Also,

$$Inverse\ document\ frequency \quad i.d.f = \ log\frac{N}{n_t}$$

where N is total no. of document in the corpus and n_t is the frequency content of the word in all the documents.

Each question is then converted to a vector where each word of it is replaced by its equivalent value from the lookup table. e.g $\overrightarrow{q_1} = (w_1, w_2, \dots\ w_n)$ and $\overrightarrow{q_2} = (w_1, w_2, \dots\ w_n)$ then finally inverse cosine of the dot product of these two vectors gives the similarity score between the two questions.

$$\theta = \cos^{-1}\frac{\overrightarrow{q_1} \cdot \overrightarrow{q_2}}{\|\overrightarrow{q_1}\|\|\overrightarrow{q_2}\|}$$

3000 pairs of questions are selected at random for testing. The sentiment value associated with each question is determined from the Microsoft cognitive services – sentiment analysis API. This is how it works: A multi-layered neural network with 3 hidden layers of 125, 25 and 5 neurons is used to tackle the task of learning to identify emotions from text using bi-gram as the text feature representation.

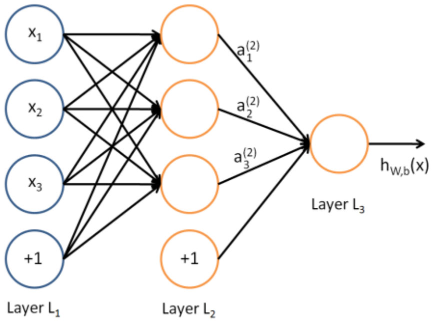

Figure 1: An example of small neural network.

In the Figure 1, circles denote the inputs to the network. The circles labelled "+1" are called bias units, and correspond to the intercept term. The leftmost layer of the network is called the input layer, and rightmost layer the output layer (which in this example has only one node). The middle layer of nodes is called the hidden layer, because its values are not observed in the training set.

The logistic regression model in R is used to predict the duplicity, based on these 3 features. Accuracy is determined by computing confusion matrix.

$$logistic\ function \quad \sigma(t) = \frac{1}{1 + e^{-t}}$$

where $t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$ and x_1, x_2 etc. are features.

4 Experiments

In this section, I conduct experiments on questions matching to observe the difference between integrated statistical and sentiment analysis against base line.

The base line for the experiment is TF-IDF method. As explained above, in this model a matching score is computed by cosine inverse of the inner product of two vectors that is formed by transforming each question into a vector.

The result derived from the Microsoft Text Analytics API is denoted Microsoft TA API. Microsoft SA API refers to the result obtained from the Microsoft's sentiment analysis API. The integrated model of TF-IDF and sentiment analysis is represented as Integrated TF-IDF and SA.

4.1 Experiment 1: Question matching

Question matching aims to determine whether two questions have the same meaning or not. Here the downloaded Quora dataset is used which contains 400,000 instances of potential duplicate pairs of questions. Experiments were run on Jupyter Notebook and R. The experimental results are listed in Table 1.

Table 1: Results on Quora Dataset

Model	Acc. (%)
TF-IDF	67
Microsoft TA API	68
Microsoft SA API	62
Integrated TF-IDF and SA	**71**

We can see that traditional model such as TF-IDF has already achieved a high accuracy of about 67% though it only uses the unigram matching signals. The integrated model of statistics and sentiment analysis performs better than TF-IDF, which indicates that sentiment associated with the asked question is also important. The best performance of my proposed model in the paper (71%) is still slightly worse than Multi-Sense word embedding (78.6%) which used neural network in its model.

5 Conclusion

In this paper, we see question matching as a special case of text matching where sentiment of the asked question plays an important role. The model proposed in the paper captures the similarity of two question using the integrated model of cosine similarity with TF-IDF weighting and sentiment analysis. Experiment results shows that the proposed model can out-perform baselines including TF-IDF.

6 Acknowledgement

Thanks, a lot to Professor Robert Platt and teaching assistant Matt Corsaro for the motivation all along. A big thanks to the anonymous reviewers for their feedback. Also, this would not have been possible without Microsoft's azure credit and access to their API.

7 References

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., & Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics, 19*(2), 263-311.

Cheng, J., & Kartsaklis, D. (2015). Syntax-aware multi-sense word embedding for deep compositional models of meaning. *arXiv preprint arXiv:1508.02354*.

Chew-Yean Yam (2015). Emotion Detection and Recognition from Text Using Deep Learning. *www.microsoft.com*

Gao, J.; Pantel, P.; Gamon, M.; He, X.; Deng, L.; and Shen, Y.2014. Modeling interestingness with deep neural networks. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*

Hardoon, D. R., & Shawe-Taylor, J. (2003). KCCA for different level precision in content-based image retrieval. In *Proceedings of Third International Workshop on Content-Based Multimedia Indexing, IRISA, Rennes, France*.

Lu, Z., & Li, H. (2013). A deep architecture for matching short texts. In *Advances in Neural Information Processing Systems* (pp. 1367-1375).

Salton, G., Fox, E. A., & Wu, H. (1983). Extended Boolean information retrieval. *Communications of the ACM*, *26*(11), 1022-1036.

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., & Pinto, D. (2014). Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computación y Sistemas*, *18*(3), 491-504.

Wu, W., Li, H., & Xu, J. (2013, February). Learning query and document similarities from click-through bipartite graph with metadata. In *Proceedings of the sixth ACM international conference on Web search and data mining* (pp. 687-696). ACM.

Xue, X., Jeon, J., & Croft, W. B. (2008, July). Retrieval models for question and answer archives. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 475-482). ACM.