UGC Approved Journal

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

Sentence Similarity Detection using Ensembling

Mukund Ananthu

Samsung, Bangalore, India

Abstract: Sentence similarity detection involves computing the degree of semantic similarity between any given pair(s) of sentences, even if the sentences do not have similar structure. Sentence similarity is an integral part of a large number of applications such as text summarization, information retrieval, text mining and artificial intelligence, to name a few. The paper introduces the concept of sentence similarity, discusses its applications and its importance. The current methods used to obtain sentence similarity are subsequently discussed. Then, an approach that uses ensembling to detect sentence similarity is proposed in the paper.

Keywords: Sentence Similarity, Ensembling, Natural Language Processing, Semantics.

I. INTRODUCTION

With the increasing ease of access to the World Wide Web, the amount of data being generated from users' interaction on the web or otherwise is growing at a rapid pace [1]. A significant portion of this data is in textual form, resulting in an increasing interest in the field of Natural Language Processing (NLP). Sentence similarity detection is one of the key problems in NLP. Sentence similarity detection deals with computation of the degree of semantic similarity between any given pair(s) of sentences. It is an important component in several applications such as text summarization, question-answer systems, information retrieval, text mining and artificial intelligence, to name a few. In text summarization, in order to convert a large document to a concise summary, it is essential to know the sentences that are similar in meaning, and hence are redundant and can be removed. Similarly, question-answer systems need to match the question entered with the questions or answers in the system. In information retrieval systems, the search query must be converted to a semantically equivalent query that the information retrieval system understands. In text mining, sentence similarity is used to extract otherwise unnoticed knowledge from text databases [2]. In artificial intelligence applications such as chat bots, where the system tries to mimic conversations with a human, it is necessary for the system to detect the similarity of the sentence typed in by the human with the sentences in its knowledge base, whose responses it knows with high degree of certainty. Given the large number of applications that rely on sentence similarity, it has become an important topic of research. However, determining the semantic equivalence of sentences is quite difficult given the large degree of variance in natural language, because of which, sentences which essentially mean the same can be constructed in a large number of ways.

II. RELATED WORK

Some of the techniques for sentence similarity use statistical methods such as Latent Semantic Analysis on a big corpus or rely on a thesaurus like WordNet [3]. Latent Semantic Analysis is an NLP technique that analyzes relationships between a collection of text documents and the terms contained in them by generating a set of concepts related to the terms and the text document [4]. WordNet is a text database containing English words grouped into sets of synonyms. These sets are linked to each other via semantic relations [5]. Word Overlap Measures is another technique that is used to determine sentence similarity. It is a combinatorial similarity measure that computes the similarity between sentence pairs based on the number of words that the two sentences have in common. Term-Frequency Inverse Document Frequency (TF-IDF) is another technique used for sentence similarity where the text is converted into vectors in vector space and similarity between two pieces of text is determined by computing the cosine similarity between the vectors representing the two texts [6]. This paper presents an approach that uses ensembling for sentence similarity detection. The ensemble consists of 4 constituent predictive models, namely a Recurrent Neural Network, a Feedforward Neural Network, a Random Forest model and a Support Vector Machine. The ensemble aggregates the predictions made by the different constituent models by taking a weighted vote from them.

III.SENTENCE SIMILARITY DETECTION USING ENSEMBLING

A. Ensembling

Ensembling involves generating a prediction that is a weighted aggregation of the predictions made by the collection of underlying constituent models. Often, the aggregated prediction made by the ensemble is more accurate than the prediction of any one single constituent model in the ensemble [7]. This is because a single model can have high

UGC Approved Journal
IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 4, Issue 8, August 2017

variability or biases that affect the accuracy of the model adversely. By combing a collection of related, but different models, one can limit the drawbacks of any individual model. Many machine learning algorithms get stuck in local optima when optimizing. This makes it difficult to find the best hypothesis, especially in scenarios where training data is small. By having a collection of predictive models in an ensemble, we can obtain a better approximation of the best hypothesis.

B. Proposed Model

In the proposed model, characteristics of the sentences such as sentiment detected in the sentences, named entities and phrases in the sentences, the cosine similarity between the sentences are extracted and used as features. In addition, a custom feature is generated, which classifies the sentences pairs into one of four categories based on the similarity/dissimilarity of the topics being discussed and the sentiments expressed towards those topics (positive, negative or neutral). These 4 categories are:

- Same topic, similar sentiment towards the topic:

Example:

Sentence 1: "The weather is excellent"
Sentence 2: "We have a really nice weather today"
Topic: Weather; Sentiment: Positive

- Same topic, different sentiment towards the topic:

Example:

Sentence 1: "The weather is good today"
Sentence 2: "We have terrible weather conditions today"
Topic: Weather; Sentiment: Positive, Negative

- Different topic, same sentiment:

Example:

Sentence 1: "The weather is superb"
Sentence 2: "The car is excellent"
Topics: Weather, Car; Sentiment: Positive

- Different topic, different sentiment:

Example:

Sentence 1: "The weather is nice"
Sentence 2: "The player did not perform well"
Topics: Weather, Player; Sentiments: Positive, Negative

These features are then used by the four constituent models and subsequently the ensemble to make a prediction on whether the sentences can be considered to be semantically equivalent or not. The constituent models used are explained in detail below:

1. A Recurrent Neural Network (RNN): RNNs are a type of neural networks where connections between units of the network form a directed cycle. They have been found to be very effective in quite a few applications, such as language modeling, text generation and machine translation. They have the ability to use their internal memory to process arbitrarily long sequences as input [8]. The RNN in the ensemble was implemented using Keras, a neural-network library written in Python [9]. A Long Short-Term Memory (LSTM) architecture was used in the RNN to handle vanishing gradient problem. Categorical Cross Entropy was used as the loss function, and the optimizer used was the Adam optimizer. The Adam optimizer is an extension to the classical stochastic gradient descent, which is used to update weights assigned to units in the network iteratively [10]. A Softmax function was used to generate the final output of the RNN. It is a normalized exponential function which is very often used in the last layer of neural network classifiers.
2. A Feedforward Neural Network: They are neural networks where layers of perceptrons are stacked together. The connections between the perceptrons are acyclic. Every perceptron in a layer is connected to the perceptrons in the next layer, resulting in unidirectional movement of information from the input layer to the output layer through hidden layers. In order to compute the error contribution of each perceptron in the neural network, backpropagation algorithm is used. The Feedforward Neural Network was implemented using PyBrain, a modular machine learning library written in Python [11].
3. A Random Forest Model: Random Forests are used for machine learning tasks such as classification and regression. They generate multiple decision trees, each of which will make a prediction for the given input, and take the mode (the prediction that appears most) of the predictions made by the decision trees. The Random Forest was implemented using Scikit-Learn, a machine learning library in Python [12].

IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified

Vol. 4, Issue 8, August 2017

4. A Support Vector Machine (SVM): A SVM is a supervised learning algorithm which can be used for both regression and classification tasks. It is a non-probabilistic linear classifier, which can also be used for non-linear classification tasks by employing kernels. It is used in a variety of applications in NLP such as spam detection, sentiment analysis and text categorization. The SVM was implemented using the Scikit-Learn machine learning library. A Radial Basis Function (RBF) kernel was used for the SVM. It is a popular kernel function that is used for SVM classification.

The above ensemble model is trained on a curated training set consisting of 200 sentence pairs that fall into different categories based on the topic being discussed and the sentiment being expressed on those topics. After training the ensemble, for every test sentence pair, the ensemble, gathers the predictions from the constituent models and predicts the sentences pairs to be similar/ dissimilar based on the number of votes for similarity/dissimilarity obtained from the constituent models.

IV. CONCLUSION

The number of users connected to the Internet is growing at a rapid rate. Consequently, the data collected on a day to day basis is also increasing at a pace that makes it impossible for humans to analyse the data manually. With advancements in technology, leading to faster and better hardware, it has become possible for humans to train machines to learn and analyse large volumes of data, being generated at a high velocity and in large variety. Given that a significant percentage of the data generated is in textual form, there is an increasing push towards developing NLP technology that can help uncover hidden insights in such large bodies of text. Detecting sentence similarity has become a crucial step in several machine learning algorithms. The multitude of ways humans can convey essentially the same message, using different sentence constructions and expressions, makes computing the semantic similarity between sentences an open problem that is difficult to solve. This paper discusses the problem of sentence similarity detection, provides an overview of the methods that have been employed so far to solve the problem and proposes an ensemble approach to tackling the same problem. Ensembling has been used with great results in quite a lot of machine learning applications, and there are statistical, computational and representational reasons for the same. It aims to reduce the error in hypothesis caused due to the limitations of a single predictive model, by having a collection of predictive models and aggregating their predictions to approximate the best hypothesis. The paper describes the ensemble approach used by the author to detect sentence similarity. The ensemble consists of four constituent models, namely, a recurrent neural network, a feedforward neural network, a random forest model and a support vector machine model. The ensemble model produced performance comparable to and in some instances better than some of the current approaches used for sentence similarity detection. Hence, it can be considered as a viable option in applications which need to determine sentence similarity.

REFERENCES

1. **"**Big Data, for better or worse: 90% of world's data generated over last two years**".** [Online]. Available: https://www.sciencedaily.com/releases/2013/05/130522085217.htm
2. Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, Keeley Crockett, "Sentence Similarity Based on Semantic Nets and Corpus Statistics", IEEE Transactions on Knowledge and Data Engineering (August 2006)
3. "UMBC Semantic Similarity Service". [Online]. Available: http://swoogle.umbc.edu/SimService/
4. "Latent Semantic Analysis". [Online]. Available: https://en.wikipedia.org/wiki/Latent_semantic_analysis
5. "WordNet". [Online]. Available: https://en.wikipedia.org/wiki/WordNet
6. Palakorn Achananuparp, Xiaohua Hu, Shen Xiajiong, "The Evaluation of Sentence Similarity Measures", International Conference on Data Warehousing and Knowledge Discovery, 2008.
7. David Opitz, Richard Maclin , "Popular Ensemble Methods: An Empirical Study", Journal of Artificial Intelligence Research 11 (1999)
8. "Recurrent Neural Network". [Online]. Available: https://en.wikipedia.org/wiki/Recurrent_neural_network
9. Keras. [Online]. Available: http://keras.io
10. Diederik P. Kingma, Jimmy Lei Ba, "ADAM: A Method for Stochastic Optimization", International Conference on Learning Representation, 2015
11. PyBrain. [Online]. Available: http://pybrain.org
12. Scikit Learn. [Online]. Available: http://scikit-learn.org
13. Victor Mijangos, Gerardo Sierra, Abel Herrera, "A Word Embeddings Model for Sentence Similarity", Research in Computing Science 117 (2016).
14. Thomas G. Dietterich, "Ensemble Methods in Machine Learning", International Workshop on Multiple Classifier Systems, 2000
15. Dipti D. Pawar, M.S. Bewoor, S.H. Patil, "Text Rank: A Novel Concept for Extraction Based Text Summarization", International Journal of Computer Science and Information Technologies, 2014
16. Quoc Le, Tomas Mikolov, "Distributed Representations of Sentences and Documents", International Conference on Machine Learning, 2014
17. Daniel Ramage, Anna N. Rafferty, Christopher D. Manning, "Random Walks for Text Semantic Similarity", Proceedings of the 2009 workshop on graph-based methods for natural language processing.
18. Yuhua Li, Zuhair Bandar, David McLean, James O'Shea, "A Method for Measuring Sentence Similarity and its Application to Conversational Agents", FLAIRS Conference, 2004
19. Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, Kilian Q. Weinberger, "From Word Embeddings To Document Distances", Proceedings of the 32nd International Conference on Machine Learning, 2015
20. Jeff Mitchell, Mirella Lapata, "Composition in Distributional Models of Semantics", Cognitive Science 34 (2010)