Machine Learning Models for Paraphrase Identification and its Applications on Plagiarism Detection

Ethan Hunt[1], Ritvik Janamsetty[1], Chanana Kinares[1], Chanel Koh[1], Alexis Sanchez[1], Felix Zhan[1]
Murat Ozdemir[2], Shabnam Waseem[2], Osman Yolcu[2]
Binay Dahal[1,2], Justin Zhan[3], Laxmi Gewali[1], Paul Oh[2]
[1]*UNITE/University of Nevada, Las Vegas*
[2]RET/University of Nevada, Las Vegas
[3] University of Arkansas

***Abstract*—Paraphrase Identification or Natural Language Sentence Matching (NLSM) is one of the important and challenging tasks in Natural Language Processing where the task is to identify if a sentence is a paraphrase of another sentence in a given pair of sentences. Paraphrase of a sentence conveys the same meaning but its structure and the sequence of words varies. It is a challenging task as it is difficult to infer the proper context about a sentence given its short length. Also, coming up with similarity metrics for the inferred context of a pair of sentences is not straightforward as well. Whereas, its applications are numerous. This work explores various machine learning algorithms to model the task and also applies different input encoding scheme. Specifically, we created the models using Logistic Regression, Support Vector Machines, and different architectures of Neural Networks. Among the compared models, as expected, Recurrent Neural Network (RNN) is best suited for our paraphrase identification task. Also, we propose that Plagiarism detection is one of the areas where Paraphrase Identification can be effectively implemented.**

***Index Terms*—Paraphrase Identification, Machine learning, Long Short Term Memory Networks, NLP**

I. Introduction

Paraphrase identification is the task of identifying if a sentence is a paraphrase of another one. It is one of the challenging tasks in Natural Language Processing. It requires representing a text in some form taking its context into consideration and formulating a metric to express the similarity between a pair of texts. The given pair of sentences or texts may look almost similar in terms of its syntactical structure but a presence of a single word or phrase may convey entirely different or opposite meanings. On the other hand, there are various applications of paraphrase identification. One of them can be automatically removing the duplicate questions in online QA forums like Quora. There are different versions of the same question in such online question-answer forums conveying the same meaning. Traditional coding would require creating billions of conditions to accurately assess whether or not two sentences are semantically the same. Another application can be the plagiarism detection task. Current applications for plagiarism essentially just check the syntax. With a quick web search, one can find many ways to circumvent plagiarism detection by switching out select words or using an article rewriter. Paraphrase identification is suggested as an application-independent framework for measuring semantic equivalence. In terms of identifying duplicate questions, according to [1], it explains that if two questions can be concluded with the same answer, both questions are semantically equivalent. The identification of semantically equivalent sentences has many applications in natural language understanding which ranges from paraphrase recognition to evaluating machine translation. There are a few challenges when it comes to processing the texts using machines. First, the computer finds it hard to recognize different words and their meanings. For example, when speaking of the companies, "Microsoft" or "Apple", the computer might mistake "Apple" as a fruit instead of a company. This problem occurs because generally, machines fail to figure out the context depicted in the text. In a given natural language sentence, there are various relationships among the words. So capturing this relationship is essential to completely understand the semantic of that sentence.

In our work, we try to perform paraphrase identification using various machine learning models and make a performance comparison among these models. In Recent years, Recurrent Neural Networks(RNNs) have proven to be very successful in machine learning tasks that relate to Natural Language. As Natural Language can be represented as a sequence of tokens(characters, words or phrases), and in general the preceding tokens affect the occurrence of next token in the sequence, RNNs, which work by taking the feedback of previous time-steps output to generate the output for subsequent time-steps, works very well. Specifically, we devise a Logistic Regression model which is the simplest machine learning model for classification task and then go on to implement a relatively more complex model: Support Vector Machines. Lastly, we will develop various models using Neural Networks including RNN model.

In the next section, we talk about some of the related works in text processing in general and also discuss the specific works done in Paraphrase Identification. Section III

978-1-7281-4607-2/19/$31.00 ©2019 IEEE
DOI 10.1109/ICBK.2019.00021

talks about our approach and the algorithms. In Section IV, we describe our experimental setups and the results. Then, a brief discussion of the results is provided in section V. Lastly, we conclude our paper with some keypoints in section VI.

II. Related Works

With an increasing amount of developing technology, further research must be done to search for solutions to improve methods of data processing using neural networks and deep learning. The process of using computers to understand Natural Language Processing (NLP) has been a challenge due to the equivocacy of texts and passages. For example, the term 'apple' can be referenced as both a company and a fruit. In this case, creating an algorithm to interpret the correct meaning would need a method to process and understand the given information. A proposed solution to this was to use semantic enrichment. To accomplish this, verbs and nouns are extracted to output a concept that represents the text [2].

In order to overcome the obstacles of natural language processing, various methods have been proposed. For instance, to combat the difficult processing of data-sparse texts, a convolutional neural network (CNN) in pair with data clustering can be used to expand text [3]. Another influential method for semantic text analysis was to use binary predicate phrases by extracting prepositional phrases [4] using PropS. If the proposition can be matched, it is presumed that the phrases are similar versions of each other.

In [5], paraphrase identification is found by using a matrix created from semantic similarities from a pair of text sections. For this method, all similarities between words were considered to improve accuracy. However, it must also be noted that this strategy was inspired by an information extraction (IE) model linking patterns to homogeneous meanings [6]. This stemmed from an (IE) model developed specifically for semantic processing [7]. For this model, matrices and vectors were formed by three elements from a clause: subject, verb, and object.

In contrast to extracting specific grammar variables, similarity can also be gleaned from language variables such as slang, syntax and lexical factors [8]. These features are then tagged and put through various N-gram models (character bigrams, trigrams, tetragrams and word unigrams, bigrams and trigrams). In [9], n-grams were used for paraphrase recognition using lexical features in combination with syntactic, composite, and semantic features. For instance, similarity can be determined by the instances of like n-grams (n-gram overlap measures), as well as dividing identical skip-grams by word variations (skip-gram overlap measures) in lexical features. N-gram overlap measures are especially similar to word co-occurrence in [10]. Another notable approach used a Siamese gated recurrent unit (GRU) neural network. Afterward, the sentences are encoded and equivalence is concluded based on an output vector [1]. It was argued that by feeding data through an additional neural network (NN) layer enhanced performance, yielding more accurate results. The method of using an additional layer can also be applied to pinpoint the location of questions in a given passage by including another gate to attention-based recurrent networks [3]. After the information from the passage is encoded with a self-matching attention mechanism, a pointer network is used to locate the positions of answers from the passage. This process was then tested on the Stanford Question Answering Dataset (SQuAD) [11].

An additional dilemma of paraphrase detection is asymmetrical paraphrases, where one sentence is more word-dense than the other [12]. Hence, it is more complicated to process in order to find the general idea or 'paraphrased' portion. A proposed solution would be to link lexical connections from each sentence in a function defined as LogSim [12]. This function aimed to be able to create a corpus with high reliability with little or no human intervention. On top of this, LogSim was designed to extract and identify paraphrases with word reorderings or syntactic differences to have semantic similarities.

Moreover, features can be classified as being either consisting of one element (primitive), or pairs of primitive elements (composite) [10]. Primitive characteristics are also used in the identification of another feature, composite features. Using composite features is beneficial because it restricts primitive features, resulting in more detailed tasks and, therefore, more specific results. [12]

In order to perform machine learning, man neural networks are needed. [13] These networks work using operators. Operators use RDF stream or data set applies a query and produces an output. Here are 3 required operators: Window, Relational, and Streaming Operators. Window Operators extract triples from an RDF stream or dataset that match a given triple pattern and are valid in a given time window. Every operator has their own job. Relational Operators work the mapping from discrete results. Streaming Operators work based on patterns to generate RDF streams from result sets. This information applies a query that is organized in a dataflow. It has a dataflow direct tree of operators, whose root nodes are a window and relational operators respectively. The routing policy decides the order in which the operators are executed at run time. CQEL's Query Engine is a specific model that was introduced. It used it own language as an extension for languages. Data Encoding was used when dealing with large amounts of data. Dictionary encoding is applied to be able to fit more data into the memory. Caching and Indexing provide faster access to data Caching is used to store intermediate results, providing faster access to the data. An index is maintained as long as it can be updated faster in order to access the data. The Operator's Routine Policy supports triple-based window operators and sliding window operators. The entire performance was evaluated in terms of average query execution time. In most things, CQELS outperforms the other approaches by orders of magnitude for it finishes 700 times faster. Usually, the operators crash if the data is too big, but EQELS performed well in efficiency and stability.

Deep learning doesn't recognize certain words yet. [2] There are certain words that a computer doesn't recognize on its

own. For example, defines “apple” or “orange” as fruit instead of different foods. The word “apple” can be defined as a company, food, or fruit. Another defect is that some verbs have different meanings. This makes it harder to understand the difference between certain phrases or short texts. Short texts lack enough context and the ambiguity that deep learning is unable to understand.

A sentence model and a similarity measurement layer make machine learning more efficient. [14]. A sentence model is used for converting a sentence into a representation for similarity measurement. A sentence model has here are many pooling types. It also multiple window sizes in the building blocks in order to learn the features of different lengths. A similarity measurement layer uses multiple similarity measurements, which compare local regions of the sentence representations from the sentence model. One way it is used to compare sentences is by flattening the sentence representations into two vectors, then use standard metrics like cosine similarity. These are the uses of a sentence model and a similarity measurement layer. Other research from the Data Science Lab includes [15]–[71], [71]–[104].

III. Our Approach

We use some commonly used approaches for preprocessing the data and encode the input into various encoding. For the Logistic Regression and SVM models, we formulate one-hot encoding and word2vec embedding before feeding the inputs to our models. First, we formulate the task of paraphrase identification as our problem definition.

A. Problem Definition

Given two sentences S_1 and S_2, where $S_1 = \langle S_1^1, S_2^1, ..., S_n^1 \rangle$ and $S_2 = \langle S_1^2, S_2^2, ..., S_n^2 \rangle$, and labels $L \in \{0, 1\}$, paraphrase identification is the task of predicting the labels: $L = 1$; if S_1 and S_2 are duplicate or $L = 0$; if they are not duplicate.

B. Preprocessing

To feed our data to the machine learning models we need to perform some preprocessing. Preprocessing includes cleaning the data and representing them in the vector form machine learning models understand. While cleaning the data, all irrelevant punctuation, uppercase letters, and symbols are either stripped or converted to ASCII character representations. Then, we create a feature vector from our input data using two schemes: One Hot Encoding and Word2Vec embedding. The algorithms for One Hot Encoding and Word2Vec are detailed along with their respective pseudocodes:

a) One Hot Encoding: One-Hot Encoding is one of the techniques to represent the input data into vector form understandable by machine learning algorithms. First, the total number of unique features are computed from the whole dataset. Then, a vector of n dimension is created for each data point with each entry in the vector specifying whether or not that data point contains that specific feature. To create a one-hot encoding of our input data, we first consider n-grams and words as a feature separately.

Algorithm 1 One Hot Encoding Algorithm

1: Create a feature set from the data set:
2: FeatureSet = $[x_1, x_2, ..., x_n]_n$ - n is the total unique characters or words
3: **for** each data point d in data set: **do**
4: **for** each feature f in d: **do**
5: d = $[d_1, d_2, ..., d_n]$, $d_i = 1$ if f is in FeatureSet
6: $d_i = 0$ if f is not in FeatureSet

Fig. 1. One Hot Encoding Algorithm

b) N-gram and Word As Feature Vector: Both N-gram and words are considered to be as a feature to construct the one hot encoding. In N-gram representation of feature vector, we consider unigram, bigram and trigrams. And separately, we consider words present in the dataset as features too.

Unigrams	Bigrams	Trigrams
Cats	**Ca**ts	**Cat**s
C**a**ts	C**at**s	C**ats**
Ca**t**s	Ca**ts**	
Cat**s**		

Fig. 2. Simple illustration of N-grams

c) Word2vec: Word2Vector is a group of word embedding models with the ability to represent raw text data while also taking context into account. This model creates vector representations of words. This model is useful to help machines understand semantic meanings in addition to word concept and context. Word2Vec algorithm includes two architectures: the continuous bag-of-words model (CBOW) and the skip-gram model [105]. In the former mentioned model, the primary function is to predict a target word based on given context words. In the latter model, its function is the obverse of the former, in that the skip-gram model attempts to predict the context words given a target word.

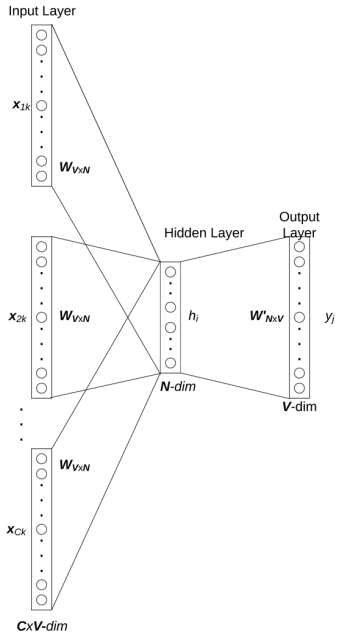

Fig. 3. CBOW Architecture

Algorithm 2 Word2Vec Algorithm

```
    Remove Punctuation
    Create Array from Text and Labels
    Labelize the Text
    Train Word2Vec model
    for words in question 1: do
        tmp = question_w 2v[w]
        vec1+ = tmp
        count+ = 1.0
    vec1/ = count
    count = 1.0
    for words in question 2: do
        tmp = question_w 2v[w]
        vec2+ = tmp
        count+ = 1.0
    vec2/ = count
    return vec1 − vec2
    Train with ML Algorithms
```

Fig. 4. Word2Vec Algorithm

C. Algorithms

A wide variety of machine learning algorithms are used in order to find the best results for paraphrase identification. All of these algorithms were used with both one hot encoding and the word2vec model.

a) Logistic Regression (LR): One algorithm used for paraphrase identification was Logistic Regression which uses a logistic function to aid in machine learning. This algorithm is most helpful for binary outputs and categorization. It is used to identify a clear comparison, or relationship, between two or more variables. Below given is a general cost function of logistic regression.

$$j(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y_i log(h_\theta(x_i) - (1-y_i) log(1-h_\theta(x_i))$$

b) Support Vector Machine (SVM): SVM is a non-probabilistic classifier of data. In other words, when a given set of data is fed through the algorithm, SVM outputs the optimum separation to categorize the information. It takes in the input of the question vector and creates a discrete classification model of the data that outputs the predicted value. In SVM, there are also multiple tuning parameters such as Kernel. Furthermore, there are various functions of Kernel, including linear, polynomial, and exponential.

c) Neural Networks (NN): The another algorithm that was used is Neural Networks (NN). Theoretically, Neural Networks are a universal function approximators. By the composition of various non-linear activations, they are able to represent any data distribution. Our neural network model comprises of a simple 3 layer deep model which takes the word2vec embedding as an input, a siamese network taking two vectors representing two sentences as an input and a Long Short Term Memory(LSTM) network.

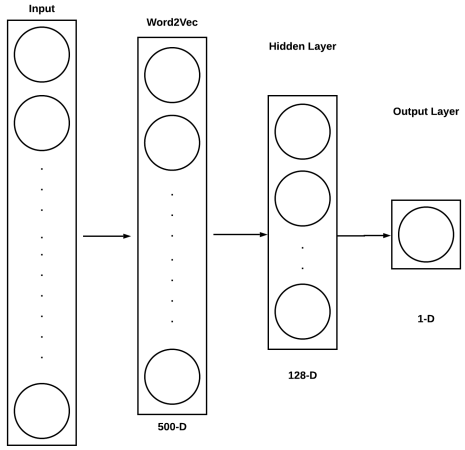

Fig. 5. A Neural Network

d) Long Short Term Memeory (LSTM): Recurrent Neural Network are the class of neural networks which takes the feedback from previous time step into consideration while making the prediction for next time step. In sequence processing task like ours, RNN can be effective. Hence, we develop a model based on LSTM, which is a variant of recurrent neural network.

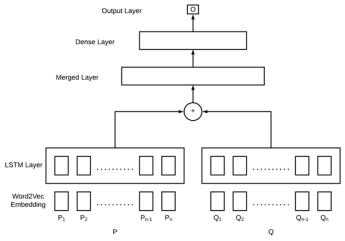

Fig. 6. Our proposed LSTM model

IV. Experimental Setup and Results

1) Dataset: The dataset used for classification purpose of our task is Quora Duplicate Question set which consisted of over 400,000 potential duplicate question pairs and a binary value indicating whether or not the questions were duplicate. The content in the questions spans a wide variety of subjects due to the public availability of the website. For training the logistic regression and SVM models, we used 50,000 data points and 10,000 were used as test data. Whereas to train the neural network models, we used 320,000 data as training data and 80,000 as test data.

id	qid1	qid2	question1	question2	is_duplicate
447	895	896	What are natural numbers?	What is a least natural number?	0
1518	3037	3038	Which pizzas are the most popularly ordered pizzas on Domino's menu?	How many calories does a Dominos pizza have?	0
3272	6542	6543	How do you start a bakery?	How can one start a bakery business?	1
3362	6722	6723	Should I learn python or Java first?	If I had to choose between learning Java and Python, what should I choose to learn first?	1

Fig. 7. A Sample of the Quora Dataset [1]

2) Experimental Setup: To prepare the data for the machine learning algorithms, we must pre-process our data accordingly. The first step in pre-processing the data consisted of separating the potential duplicate question sets into question one, question two, and a binary value which represents whether or not the question set is duplicate. Next, everything in the dataset was normalized to lowercase as this would increase the performance of the machine learning algorithms. For paraphrase identification, special care must be taken in the selection of the feature set as this can have a profound impact on the performance of our machine learning algorithms.

A. One Hot Encoding

The first step in this process was obtaining all of the unique character n-grams from the dataset. This included all of the unique unigrams, bigrams, and trigrams which were then added to the feature set. The next step for one hot encoding was the vector creation process which includes creating two lists, one for each question set, and appending a binary value to each of those lists based on if an n-gram from our feature set matches something in question one or question two. It is after the vector creation process where our data is finally ready to be split into both testing and training data.

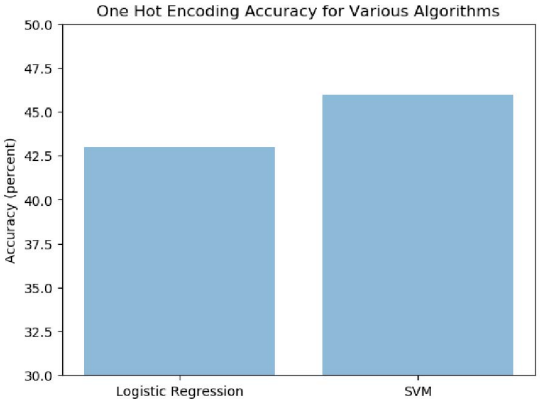

Fig. 8. Results for One Hot Encoding

B. Words as a Feature Vector

First, the questions from our dataset were filtered to remove certain punctuation that was believed to negatively impact our algorithms. After this, the vector creation process occurs which largely remains the same as the process used for n-grams.

C. Word2Vec

For the pre-processing for the word2vec model, our data must be converted into LabeledSentence objects and then the vector dimension space must be defined. The vocabulary is then created and the word2vec is trained. After this, the vector creation process ensues. We used 500 dimension hidden layer for word2vec model which is embedded later as an input representation.

D. Results From Algorithm

a) Logistic Regression (LR) and SVM models: Figure 8 and 9 shows the result for Logistic Regression and SVM models using one-hot encoding and word2vec embedding. The result shows word2vec performs significantly better than one-hot encoding scheme and SVM performs slightly better than logistic regression.

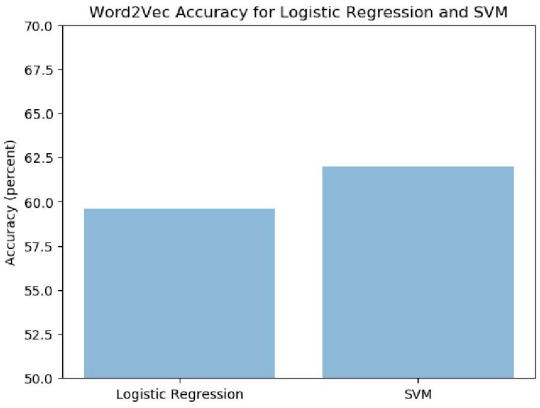

Fig. 9. Accuracy Results for Logistic Regression and SVM

b) Neural Networks (NN): All of the neural network models are fed with 500 dimensional word2vec embedding. The hidden layer size used was 128 units with a batch size of 32 with the model being trained over 15 epochs. This technique was used using the word2vec model for just words.

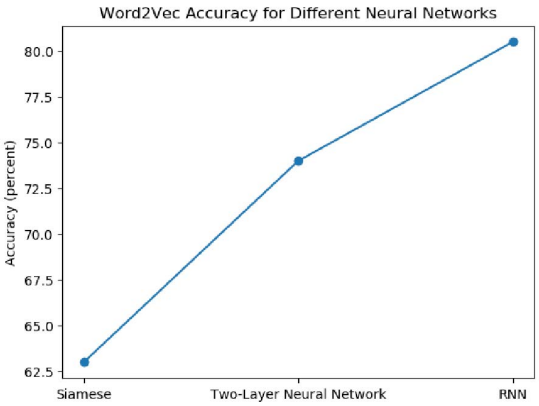

Fig. 10. Accuracy Results for Different Neural Networks

V. Discussions

Various experiments were performed using different models. The lowest accuracy obtained was for logistic regression using a one-hot encoding. The accuracy was around 0.43 while SVM performed slightly better at 0.45. The sub-par result was expected as the one-hot encoding is a sparse representation of input and when the feature space is large the input vector becomes very sparse and it can't be processed effectively. In our case, using N-grams as features, the feature space size was 1300 and using words as features, the feature space size was around 80000. Hence, the result is attributed to that. Next, we represented input as word2vec embedding and the result from the same two models was significantly improved. Word2vec generates a dense representation of input taking all the context of a given text into consideration. The accuracy for two models was around 0.6 and 0.62. The result from various models using neural networks is shown in figure 10. The highest accuracy obtained is for the LSTM model. As expected, LSTM model performed better than other variants of neural networks as they are more suited for sequence data.

VI. Conclusion

Paraphrase identification can be used in many applications. One of them, we propose in Plagiarism detection. One of the main problems with plagiarism checkers is they check for syntactical structures only instead of semantic meaning. Our model can be used to develop a plagiarism detection system where a simple rewrite of a text will be flagged as plagiarized. We have developed a simple application using our model for this purpose. In conclusion, Various machine learning algorithms were implemented for paraphrase identification tasks. Specifically, we used Logistic Regression, Support Vector Machine, and Neural Networks. As expected, recurrent neural networks (RNN) were found to produce the most accurate results. Furthermore, we propose that Paraphrase Identification can be implemented for plagiarism detection effectively and also developed a simple application for the demonstration purpose.

Acknowledgment

This research was supported in part by the Department of Defense under the Army Educational Outreach Program (AEOP) and the National Science Foundation under the Research Experiences for Teachers (RET) program.

References

[1] Y. Homma, S. Sy, and C. Yeh, "Detecting duplicate questions with deep learning," in *Proceedings of the International Conference on Neural Information Processing Systems (NIPS*, 2016.

[2] J. Zhan and B. Dahal, "Using deep learning for short text understanding," *Journal of Big Data*, vol. 4, no. 1, p. 34, 2017.

[3] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 189–198.

[4] V. Shwartz, G. Stanovsky, and I. Dagan, "Acquiring predicate paraphrases from news tweets," in *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, 2017, pp. 155–160.

[5] S. Fernando and M. Stevenson, "A semantic similarity approach to paraphrase detection," in *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*, 2008, pp. 45–52.

[6] M. Stevenson and M. A. Greenwood, "A semantic approach to ie pattern induction," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 379–386.

[7] R. Yangarber, "Counter-training in discovery of semantic patterns," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 343–350.

[8] C. Peersman, W. Daelemans, and L. Van Vaerenbergh, "Predicting age and gender in online social networks," in *Proceedings of the 3rd international workshop on Search and mining user-generated contents*. ACM, 2011, pp. 37–44.

[9] A. Rajkumar and A. Chitra, "Paraphrase recognition using neural network classification," *International Journal of Computer Applications*, vol. 1, no. 29, pp. 42–47, 2010.

[10] V. Hatzlvassiloglou, J. L. Klavans, and E. Eskin, "Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning," in *1999 Joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, 1999.

[11] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[12] C. Joao, D. Gaël, and B. Pavel, "New functions for unsupervised asymmetrical paraphrase detection," *Journal of Software*, vol. 2, no. 4, pp. 12–23, 2007.

[13] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, "Semantic enrichment of twitter posts for user profile construction on the social web," in *Extended semantic web conference*. Springer, 2011, pp. 375–389.

[14] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1576–1586.

[15] N. R. R. M. M. S. M. B. J. Z. L. G. P. O. Felix Zhan, Anthony Martinez, "Beyond cumulative sum charting in non-stationarity detection and estimation," *IEEE Access*, 2019.

[16] Z. J. Schwob, M. and D. A., "Modeling cell communication with time-dependent signaling hypergraphs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, p. doi: 10.1109/TCBB.2019.2937033, 2019.

[17] C. Chiu and J. Zhan, "Deep learning for link prediction in dynamic networks using weak estimators," *IEEE Access*, vol. 6, no. 1, pp. 35 937 – 35 945, 2018.

[18] M. Bhaduri and J. Zhan, "Using empirical recurrences rates ratio for time series data similarity," *IEEE Access*, vol. 6, no. 1, pp. 30 855–30 864, 2018.

[19] J. Wu, J. Zhan, and S. Chobe, "Mining association rules for low frequency itemsets," *PLOS ONE*, vol. 13, no. 7, 2018.

[20] P. Ezatpoor, J. Zhan, J. Wu, and C. Chiu, "Finding top-k dominance on incomplete big data using mapreduce framework," *IEEE Access*, vol. 6, no. 1, pp. 7872–7887, 2018.

[21] P. Chopade and J. Zhan, "Towards a framework for community detection in large networks using game-theoretic modeling," *IEEE Transactions on Big Data*, vol. 3, no. 3, pp. 276–288, 2017.

[22] M. Bhaduri, J. Zhan, and C. Chiu, "A weak estimator for dynamic systems," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 27 354–27 365, 2017.

[23] M. Pirouz and J. Zhan, "Toward efficient hub-less real time personalized pagerank," *IEEE Transactions on Big Data*, vol. 5, no. 1, pp. 26 364–26 375, 2017.

[24] M. Bhaduri, J. Zhan, C. Chiu, and F. Zhan, "A novel online and non-parametric approach for drift detection in big data," *IEEE Access*, vol. 5, no. 1, pp. 15 883–15 892, 2017.

[25] C. Chiu, J. Zhan, and F. Zhan, "Uncovering suspicious activity from partially paired and incomplete multimodal data," *IEEE Access*, vol. 5, no. 1, pp. 13 689 – 13 698, 2017.

[26] R. Ahn and J. Zhan, "Using proxies for node immunization identification on large graphs," *IEEE Access*, vol. 5, no. 1, pp. 13 046–13 053, 2017.

[27] M. Wu, J. Zhan, and J. Lin, "Ant colony system sanitization approach to hiding sensitive itemsets," *IEEE Access*, vol. 5, no. 1, pp. 10 024–10 039, 2017.

[28] J. Zhan and B. Dahal, "Using deep learning for short text understanding," *Journal of Big Data*, vol. 4, no. 34, pp. 1–15, 2017.

[29] J. Zhan, S. Gurung, and S. P. K. Parsa, "Identification of top-k nodes in large networks using katz centrality," *Journal of Big Data*, vol. 4, no. 16, 2017.

[30] J. Zhan, T. Rafalski, G. Stashkevich, and E. Verenich, "Vaccination allocation in large dynamic networks," *Journal of Big Data*, vol. 4, no. 2, pp. 161–172, 2017.

[31] J. M.-T. Wu, J. Zhan, and J. C.-W. Lin, "An aco-based approach to mine high-utility itemsets," *Knowledge-Based Systems*, vol. 116, pp. 102–113, 2017.

[32] M. Pirouz, J. Zhan, and S. Tayeb, "An optimized approach for community detection and ranking," *Journal of Big Data*, vol. 3, no. 22, pp. 102–113, 2017.

[33] J. Zhan, V. Gudibande, and S. P. K. Parsa, "Idenfication of top-k influential communities in large networks," *Journal of Big Data*, vol. 3, no. 16, 2016.

[34] M. Pirouz and J. Zhan, "Node reduction in personalized page rank estimation for large graphs," *Journal of Big Data*, vol. 3, no. 12, 2016.

[35] H. Selim and J. Zhan, "Towards shortest path identification on large networks," *Journal of Big Data*, vol. 3, no. 10, 2016.

[36] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 5, pp. 1–14, 2015.

[37] P. Chopade and J. Zhan, "Structural and functional analytics for community detection in large-scale complex networks," *Journal of Big Data*, vol. 2, no. 1, pp. 1–28, 2015.

[38] J. Zhan and X. Fang, "A computational framework for detecting malicious actors in communities," *International Journal of Privacy, Security, and Integrity*, vol. 2, no. 1, pp. 1–20, 2014.

[39] A. Rajasekar, H. Kum, M. Cross, J. Crabtree, S. Sankaran, H. Lander, T. Carsey, G. King, and J. Zhan, "The databridge," *Science Journal*, vol. 2, no. 1, pp. 1–14, 2013.

[40] J. Zhan, X. Fang, and N. Koceja, "A novel framework on data reduction," *Science Journal*, vol. 2, no. 1, pp. 15–23, 2013.

[41] A. Doyal and J. Zhan, "Towards ddos defense and traceback," *International Journal of Privacy, Security, and Integrity*, vol. 1, no. 4, pp. 299–311, 2013.

[42] J. Zhan and X. Fang, "Towards social network evolution," *Human Journal*, vol. 1, no. 1, pp. 218–233, 2012.

[43] J. Zhan, J. Oommen, and J. Crisostomo, "Anomaly detection in dynamic systems using weak estimator," *ACM Transaction on Internet Technology*, vol. 11, no. 1, pp. 53–69, 2011.

[44] J. Zhan and X. Fang, "Social computing: The state of the art," *International Journal of Social Computing and Cyber-Physical Systems*, vol. 1, no. 1, pp. 1–12, 2011.

[45] N. Mead, M. S., and J. Zhan, "Integrating privacy requirements considerations into a security requirements engineering method and tool," *International Journal of Information Privacy, Security and Integrity*, vol. 1, no. 1, pp. 106–126, 2011.

[46] J. Zhan, "Granular computing in privacy-preserving data mining," *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, vol. 1, no. 3, pp. 272–288, 2010.

[47] J. Wang, J. Zhang, and J. Zhan, "Towards real-time performance of data privacy protection," *International Journal of Granular Computing, Rough Sets and Intelligent Systems*, vol. 1, no. 4, pp. 329–342, 2010.

[48] J. Zhan, "Secure collaborative social networks," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 6, pp. 682–689, 2010.

[49] J. Zhan, H. C., I. Wang, T. Hsu, C. Liau, and W. D., "Privacy-preserving collaborative recommender systems," *IEEE Transaction on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 4, pp. 472–476, 2010.

[50] H. Park, J. Hong, J. Park, J. Zhan, and D. Lee, "Attribute-based access control using combined authentication technologies," *IEEE Transaction on Mobile Computing*, vol. 9, no. 6, pp. 824–837, 2010.

[51] I. Wang, C. Shen, J. Zhan, T. Hsu, C. Liau, and D. Wang, "Empirical evaluations of secure scalar product," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 39, no. 4, pp. 440–447, 2009.

[52] A. Inoue, T. Wong, and J. Zhan, "Applications of machine learning to information security and privacy," *Journal of Japanese Society for Fuzzy Theory and Intelligent Informatics*, vol. 19, no. 3, pp. 222–232, 2009.

[53] J. Zhan, "Privacy-preserving collaborative data mining," *IEEE Computational Intelligence Magazine*, vol. 3, no. 2, pp. 31–41, 2008.

[54] A. Bashir and J. Zhan, "Not always a blunt tool – legislation in the context of privacy externalities," *Communications of the Chinese Cryptology and Information Security Association*, vol. 2, no. 1, pp. 36–48, 2008.

[55] J. Zhan and V. Rajamani, "The economic aspects of privacy," *International Journal of Security and Its Applications*, vol. 2, no. 3, pp. 101–108, 2008.

[56] J. Zhan, "The economic aspects of privacy," *International Journal of Security and Its Applications*, vol. 2, no. 3, pp. 101–108, 2008.

[57] W. Zhang, P. Wang, K. Peace, J. Zhan, and Y. Zhang, "On truth, uncertainty, and bipolar logic," *Journal of New Mathematics and Natural Computing*, vol. 4, no. 2, pp. 55–65, 2008.

[58] N. Mead, V. Viswanathan, and J. Zhan, "Incorporating security requirements engineering into standard lifecycle processes," *International Journal of Security and Its Applications*, vol. 2, no. 4, pp. 67–80, 2008.

[59] J. Zhan and S. Matwin, "Privacy-preserving data mining in electronic surveys," *International Journal of Network Security*, vol. 4, no. 3, pp. 318–327, 2007.

[60] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving multi-party decision tree induction," *International Journal of Business Intelligence and Data Mining*, vol. 2, no. 2, pp. 197–212, 2007.

[61] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving collaborative association rule mining," *Journal of Network and Computer Applications*, vol. 30, no. *, pp. 1216–1227, 2007.

[62] J. Zhan, L. Chang, and S. Matwin, "Building k-nearest neighbor classifiers on vertically partitioned private data," *International Journal of Network Security*, vol. 1, no. 1, pp. 46–51, 2005.

[63] J. Zhan and S. Matwin, "Privacy preserving support vector machine classification," *International Journal of Intelligent Information and Database Systems*, vol. 1, no. 3/4, pp. 356–385, 2005.

[64] S. Matwin, L. Chang, R. Wright, and J. Zhan, "Editorial - "privacy and security aspects of data mining" special issue," *International Journal of Information and Computer Security*, vol. 2, no. 1, p. 1, 2005.

[65] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving electronic voting," *International Journal of Information and Security*, vol. 15, no. 2, pp. 165–180, 2004.

[66] L. Singh and J. Zhan, "Measuring topological anonymity in social networks," in *Proceedings of the IEEE International Conference on Granular Computing*, Silicon Valley, USA, November 2007, pp. 770–774.

[67] C. Barron, H. Yu, and J. Zhan, "Cloud computing security case studies and research," in *2013 International Conference of Parallel and Distributed Computing*, London, UK, July 2013.

[68] H. Park, B. Kim, D. Lee, Y. Chung, and J. Zhan, "Secure similarity search," in *Proceedings of the IEEE International Conference on Granular Computing*, Silicon Valley, USA, November 2007, pp. 598–598.

[69] S. Miyazaki, N. Mead, and J. Zhan, "Computer-aided privacy requirements elicitation techniques," in *Proceedings of the IEEE Asia Pacific International Conference on Services Computing*, Yilan, Taiwan, December 2008, pp. 367–372.

[70] X. Fang and J. Zhan, "Online banking authentication using mobile phones," in *Proceedings of the International Symposium on Financial Security*. Busan, Korea: IEEE CS Press, May 2010.

[71] J. Zhan and L. Chang, "Privacy-preserving collaborative sequential pattern mining with horizontally partitioned datasets," in *Proceedings of the International Conference on Data Privacy and Security in A Global Society*, Skiathos, Greece, May 2004, pp. 242–252.

[72] T. Yu, D. Lee, and J. Zhan, "Multi-party k-means clustering with privacy consideration," in *Proceedings of IEEE International Symposium on Parallel and Distributed Processing with Applications*, Taipei, Taiwan, September 2010.

[73] Y. Duan, J. Canny, and J. Zhan, "Efficient privacy-preserving association rule mining: P4p style," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, Honolulu, Hawaii, USA, April 2007, pp. 654–660.

[74] J. Zhan and X. Fang, "A novel trust computing system for social networks," in *IEEE International Conference on Privacy, Security, Risk and Trust*. MIT, Boston, USA: IEEE CS Press, October 2011.

[75] J. Zhan and V. Rajamani, "The economics of privacy: People, policy and technology," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 579–584.

[76] G. Blosser and J. Zhan, "Privacy-preserving collaborative social networks," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 543–548.

[77] K. Prakobphol and J. Zhan, "A novel outlier detection scheme for network intrusion detection systems," in *Proceedings of the International Conference on Information Security and Assurance*. IEEE CS Press, April 2008, pp. 555–560.

[78] F. Zhan, G. Laines, S. Deniz, S. Paliskara, I. Ochoa, I. Guerra, M. Pirouz, C. Chiu, S. Tayeb, E. Ploutz, J. Zhan, L. Gewali, and P. Oh, "An efficient alternative to personalized page rank for friend recommendations," in *IEEE Consumer Communications & Networking Conference*, Las Vegas, USA, January 2018.

[79] F. Zhan, G. Laines, S. Deniz, S. Paliskara, I. Ochoa, I. Guerra, S. Tayeb, C. Chiu, M. Pirouz, E. Ploutz, L. G. Justin Zhan and, and P. Oh, "Prediction of online social networks users' behaviors with a game theoretic approach," in *IEEE Consumer Communications & Networking Conference*, Las Vegas, USA, January 2018.

[80] J. Zhan and L. Chang, "Privacy-preserving data mining," in *Proceedings of the IEEE International Conference on Data Mining, Workshop of Foundations and New Directions in Data Mining*, Melbourne, Florida, USA, November 2003, pp. 65–71.

[81] J. Zhan, L. Chang, and S. Matwin, "Bayesian network induction with incomplete private data," in *Proceedings of the International Conference on Electronic Business*, Beijing, China, December 2004, pp. 1119–1124.

[82] J. Zhan, S. Matwin, N. Japkowicz, and L. Chang, "Association rule mining and privacy protection," in *Proceedings of the International Conference on Electronic Business*, Beijing, China, December 2004, pp. 1172–1178.

[83] J. Zhan and S. Matwin, "Privacy-preserving electronic surveys," in *Proceedings of the International Conference on Electronic Business*, Beijing, China, December 2004, pp. 1179–1185.

[84] J. Zhan, L. Chang, and S. Matwin, "Collaborative data mining and privacy protection," in *Foundation and Novel Approach in Data Mining, Edited by T.Y. Lin, S. Ohsuga, C.J. Liau, and X. Hu*. Springer-Verlag, August 2004, pp. 213–227.

[85] ——, "Privacy-preserving multi-party decision tree classification," in *Proceedings of the 2004 Annual IFIP WG 11.3 Working Conference on Data and Application Security*, Sitges, Catalonia, Spain, July 2004, pp. 341–355.

[86] ——, "Privacy-preserving collaborative sequential pattern mining," in *Proceedings of the SIAM International Conference on Data Mining, Workshop on Link Analysis, Counter-terrorism, and Privacy*, Lake Buena Vista, Florida, April 2004, pp. 61–72.

[87] ——, "Privacy-preserving naive bayesian classification," in *Proceedings of the IASTED International Conference on Artificial Intelligence and Applications*, Innsbruck, Austria, February 2004, pp. 141–155.

[88] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving association rule mining," in *Proceedings of the 2005 Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Storrs, Connecticut, USA, August 2005, pp. 153–165.

[89] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving support vector machine learning," in *Proceedings of the International Conference on Electronic Business*, Hong Kong, December 2005.

[90] J. Zhan, M. S., and L. Chang, "Privacy-preserving decision tree classification over horizontally partitioned data," in *Proceedings of the International Conference on Electronic Business*, Hong Kong, December 2005.

[91] J. Zhan, L. Chang, and S. Matwin, "Privacy-preserving sequential pattern mining over vertically partitioned data," in *Proceedings of the International Conference on Electronic Business*, Hong Kong, December 2005.

[92] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving naïve bayesian classification over horizontally partitioned data," in *Proceedings of the International Conference on Electronic Business*, Hong Kong, December 2005.

[93] J. Zhan and S. Matwin, "Privacy and security issues in medical informatics," in *the Electronic Health Information and Privacy Conference*, Ottawa, Canada, November 2005.

[94] J. Zhan, L. Chang, and S. Matwin, "How to prevent private data from being disclosed to a malicious attacker," in *Proceedings of the IEEE International Conference on Data Mining Workshop on Foundations of Semantic Oriented Data and Web Mining*, Houston, Texas, USA, November 2005, pp. 41–46.

[95] ——, "Privacy-preserving naïve bayesian classification over vertically partitioned data," in *Proceedings of the IEEE International Conference on Data Mining Workshop on Foundations of Semantic Oriented Data and Web Mining*, Houston, Texas, USA, November 2005, pp. 47–53.

[96] J. Zhan, S. Matwin, and L. Chang, "Multi-party sequential pattern mining over private data," in *Proceedings of the IEEE International Conference on Data Mining Workshop on Multi-Agent Data Warehousing and Multi-Agent Data Mining*, Houston, Texas, USA, November 2005, pp. 112–120.

[97] ——, "Privacy-preserving decision tree classification over vertically partitioned data," in *Proceedings of the IEEE International Conference on Data Mining Workshop on Multi-Agent Data Warehousing and Multi-Agent Data Mining*, Houston, Texas, USA, November 2005, pp. 121–129.

[98] J. Zhan, L. Chang, and S. Matwin, "Building k-nearest neighbor classification on vertically partitioned private data," in *Proceedings of the IEEE International Conference on Granular Computing*, Beijing, China, July 2005, pp. 708–711.

[99] J. Zhan, S. Matwin, and L. Chang, "Privacy-preserving clustering over horizontally partitioned data," in *Proceedings of Artificial Intelligence Studies: VII International Conference on Artificial Intelligence AI-20'2005*, Poland, June 2005, pp. 39–48.

[100] J. Zhan, L. Chang, and S. Matwin, "How to construct support vector machines without breaching privacy," in *Proceedings of Artificial Intelligence Studies: VII International Conference on Artificial Intelligence AI-20'2005*, Poland, June 2005, pp. 49–58.

[101] J. Zhan, "Research directions in data mining and privacy," in *International Conference on Services Systems and Services Management*, Chongqing, China, June 2005, pp. 49–58.

[102] J. Zhan, S. Matwin, and L. Chang, "Private mining of association rules," in *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, Atlanta, Georgia, May 2005, pp. 72–80.

[103] J. Zhan, L. Chang, and S. Matwin, "Collaborative association rule mining by sharing private data," in *Proceedings of the Montreal Conference On E-Technologies*, Montreal, Canada, January 2005, pp. 193–197.

[104] J. Zhan and S. Matwin, "A crypto-approach to privacy-preserving data mining," in *Proceedings of the IEEE International Conference on Data Mining Workshop on Privacy Aspect of Data Mining*, Hong Kong, December 2006, pp. 546–550.

[105] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.