A Distance Approach for Open Information Extraction Based on Word Vector

Liu Peiqian[1,2], Wang Xiaojie[1]
[1] School of Computer Science, Beijing University of Posts and Telecommunications
Beijing 100876 - China
[e-mail: liupeiqian@sina.com.cn; xjwang@bupt.edu.cn]
[2] School of Computer Science, Henan Polytechnic University
Henan jiaozuo 454003 - China
[e-mail: liupeiqian@hpu.edu.cn]
*Corresponding author: Liu Peiqian

Received May 29, 2017; revised November 7, 2017; revised December 28, 2017; accepted January 30, 2018; published June 30, 2018

Abstract

Web-scale open information extraction (Open IE) plays an important role in NLP tasks like acquiring common-sense knowledge, learning selectional preferences and automatic text understanding. A large number of Open IE approaches have been proposed in the last decade, and the majority of these approaches are based on supervised learning or dependency parsing. In this paper, we present a novel method for web scale open information extraction, which employs cosine distance based on Google word vector as the confidence score of the extraction. The proposed method is a purely unsupervised learning algorithm without requiring any hand-labeled training data or dependency parse features. We also present the mathematically rigorous proof for the new method with Bayes Inference and Artificial Neural Network theory. It turns out that the proposed algorithm is equivalent to Maximum Likelihood Estimation of the joint probability distribution over the elements of the candidate extraction. The proof itself also theoretically suggests a typical usage of word vector for other NLP tasks. Experiments show that the distance-based method leads to further improvements over the newly presented Open IE systems on three benchmark datasets, in terms of effectiveness and efficiency.

Keywords: open information extraction, word vector, Maximum Likelihood Estimation, Bayes Inference, natural language processing

This research was supported by a research grant from the National Natural Science Foundation of China [No. 61273365].

http://doi.org/10.3837/tiis.2018.06.003 **ISSN : 1976-7277**

1. Introduction

Web-scale information extraction has attracted considerable attention in recent years. Typically, Information Extraction (IE) systems train an extractor for each target relation from hand-labeled training examples. This approach to IE cannot scale to corpora with arbitrary target relations, or with a very large amount of target relations. Through automatically identifying relation phrases in English sentences, the Open IE systems can extract arbitrary relations from sentences, without requiring any pre-specified vocabulary. Open IE prefers speed to deeper processing, which aids in scaling to Web size corpora. Preemptive Information Extraction and Open Information Extraction are the first paradigms that relax the restriction of a given vocabulary of relations and scale to all relation phrases expressed in text [1]. Preemptive IE relies on document and entity clustering, which is too costly for Web-scale IE.

Ever since 2003, the KnowItAll project at the University of Washington has started to extract high-quality extracions from massive Web corpora. In 2007, they proposed the Open Information Extraction (Open IE) paradigm, TEXTRUNNER, which aims to scale IE methods to the size and diversity of the Web corpus [2]. TEXTRUNNER used a Naive Bayes model with unlexicalized POS and NP-chunk features, trained on examples heuristically generated from the Penn Treebank. Subsequent work showed that utilizing a linear-chain CRF [3] or Markov Logic Network [4] can result in improved extractions. The WOE system makes use of Wikipedia as a source of training data for their extractors, which results in further improvements over TEXTRUNNER [5]. They also show that dependency parse features lead to a dramatic increase in precision and recall over shallow linguistic features，but at the cost of extraction speed. Similar systems are OLLIE [6], NELL [7], Wanderlust [8], SOFIE [9], Prospera [10], PATTY [11], Sonex [12] and Exemplar [13].

There are two significant problems in all prior Open IE systems: incoherent extractions and uninformative extractions. Incoherent extractions are cases where the extracted relation phrase has no meaningful interpretation. Incoherent extractions arise because the learned extractor makes a sequence of decisions about whether to include each word in the relation phrase, often resulting in incomprehensible relation phrases.

In response to these limitations, REVERB introduces a novel open extractor based on two constraints. REVERB first identifies relation phrases that satisfy the syntactic and lexical constraints, and then finds a pair of NP arguments for each identified relation phrase. The resulting extractions are then assigned a confidence score using a logistic regression classifier trained on labeled data. Since there is no large training set available for Open IE, the confidence score from the classifier is not so satisfied. Another problem in REVERB (and other systems) is the failure of relation extraction process due to the complex structure of the input sentence. In fact, a relation between entity pairs in simple sentences is detected easier by the Open IE systems rather than in complex sentences.

Text simplification has been employed in relation extraction task as in [14] and [15]. As for Open IE system, the simpler clause extraction has been proposed in [16] [17] [18]. In the simplification process, manually defined rules or pattern learning are utilized. Early researches in text simplification have considered punctuation marks such as comma, semicolon, and parentheses as important markers to split the clauses [19]. An extensive research of relative clause extraction on simplifying text is described in [20].

The recently proposed PCEOE system [21] outperforms some other systems through clause extraction from the input sentences. The clause extraction procedure is based on manually defined rules and dependency parsing, so the computational cost is much higher than

REVERB. A similar system is ArgOE [22], which is a rule-based system and relies heavily on dependency parse features.

Another solution for complex sentences is the SRL-based systems. Due to their deeper parsing and Semantic Role Labeling (SRL), these systems can handle complex syntax and long-range dependencies. The first two SRL-based systems are SRL-IE-UIUC and SRL-IE-Lund [23]. Recently, Mausan proposed a publicly available SRL-based system, OpenIE4 [24], which obtains better performance of precision and recall, but at the cost of efficient, since deep syntactic analysis is much computationally expensive.

In this article, we use distance among the elements as the confidence score of the candidate extraction. The distance is computed based on Google word vector [25], which is a deep learning algorithm without requiring hand-labeled training corpus. Experiments show that the distance-based method leads to further improvements over the newly presented Open IE systems. The main contributions of this paper are as follows.

(1) A novel method is presented for web scale open information extraction based on Google word vector.

(2) The new method is rigorously proved in mathematics via Bayes theory and Artificial Neural Network theory. The proveness itself also suggests theoretically a typical usage of word vector for other NLP tasks.

(3) Performance comparisons with some newly proposed methods on three datasets are presented. Experiments show that the new algorithm achieves better performance compared to some recently presented systems, in terms of effectiveness and efficiency.

The rest of this paper is organized as follows. In section 2, we review briefly the newly presented Open IE systems, and then the novel architecture for computing word vectors proposed by Google researchers. Section 3.1 describes our distance-based approach for evaluating candidate extraction. The new approach is proved to be an equivalence to Maximum Likelihood Estimation in section 3.2 by means of Bayesian analysis. Section 4 presents experimental evaluation of our new approach, compared with the newly presented Open IE systems.

2. Related Work

REVERB is the second generation of open information system. Given a POS-tagged and NP-chunked sentence s as input, the algorithm returns an extraction triple as following steps:

(i). Relation extraction: find all verbs in sentence s, and for each v in the sentence the algorithm produces a relation phrase r_v respectively. The relation phrases must satisfy the lexical constraint and syntactic constraint as shown in **Fig. 1**.

V | *VP* | *VW*P*
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. Marker)

Fig. 1. A regular expression for the syntactic constraint on relation phrases.

(ii). Argument extraction: find the nearest noun phrases x and y for each identified r_v in step1. Where x is on the left side of r_v and y is on the right side of r_v.

(iii). A logistic regression classifier is trained on a hand-labeled set of random Web sentences, and the classifier's weights are used to order the extractions.

REVERB focuses on identifying a more meaningful and informative relation phrase and outperforms the previous Open IE systems by significant margins, but it often fails to extract relations from sentences with complex structures.

In 2015, Ade Romadhony et al. proposed a text simplification approach for Open IE system through clause extraction procedure (PCEOE), resulting in performance improvement over some other Open IE systems. Text simplification is a process which can reduce the language complexity while the original information and its meaning are still preserved [26]. The benefits of text simplification vary from reading comprehension improvement for human to the ease of processing for computer. Several domains have exploited the text simplification such as machine translation, information retrieval, information extraction, and text summarization .

The proposed clause extraction approach consists of two steps. The first step simplifies the mentioned entity starting with determiner, and will detect a list of entity pairs from a sentence. The second step produces certain part of input sentences that contain information about entity pairs and relations between them. The experimental results show that the performance of Open IE system increases by extracting simpler clauses, compared with previous work [27].

ArgOE includes also two steps. In its first step, argument structures are detected based on the standard CoNLL-X dependency parser. Each argument structure is the abstract representation of a clause. For each argument structure, ArgOE generates a set of triples in its second step, based on a set of simple rules. The experimental results show that ArgOE performs better than some learning-based systems, in terms of precision and recall, as reported in [22].

Although semantic role labeling and Open IE are developed mostly in isolation, but they are quite related to each other. The most recent SRL-based Open IE system is OpenIE4, which relies on dependency parsing and a classifier trained over PropBank. OpenIE4 applies a pipeline of parsing, argument identification, and classification for relation extraction. In evaluation, OpenIE4 obtains a good balance of precision and recall. We could expect that OpenIE4 has better precision and recall compared to REVERB, but it is much slower than REVERB.

All of the previous Open IE systems can be divided in two categories: those systems requiring training data to learn a classifier and those based on manually defined rules. In addition, each category can be organized in two sub types: those based on shallow syntactic analysis and those utilizing dependency parsing trees. To sum up, there are four categories of Open IE systems:

(1) Learning-based and dependency parsing: e.g. WOE, OLLIE and OpenIE4.

(2) Learning-based and shallow syntax: e.g. TEXTRUNNER, StatSnowball.

(3) Rule-based and dependency parsing: e.g. ArgOE and PCEOE.

(4) Rule-based and shallow syntax: e.g. REVERB,

Our system belongs to the fourth category and quite similar to REVERB. The main difference between them is that the two systems use different methods to calculate confidence scores for the candidate extractions.

We should discuss here the merits and demerits of the newly presented methods. A more detailed discussion on the previous Open IE systems was presented in [28]. REVERB extracts relations under lexical constraints and syntactic constraints resulting in more meaningful and informative relation phrases. Since REVERB depends only on shallow parsing, it is quite efficient than PCEOE, ArgOE and OpenIE4, which are based on dependency parsing. Nevertheless, shallow parsing cannot deal with complex sentences effectively. While

dependency parsing outperforms shallow parsing in terms of precision, but it is too costly for Web-scale IE. Although the text simplification procedure of PCEOE improves the precision, the recall decreases, since many candidate extractions may be filtered out during the text simplification procedure. In addition, since OpenIE4 is a learning-based system, its performance suffers from the lack of training data.

REVERB employs a logistic regression classifier to evaluate the candidate extractions, but due to the lack of training data, performance of the classifier is restricted. To solve this problem, we present a new method which uses the distances among x, y *and* r_v as confidence score of the candidate extraction (x, r_v, y). The distance value is based on Google word2vec, which is an unsupervised learning algorithm without the use of hand-labeled training data. Since unlabeled data can be easily acquired on the web, the proposed method could achieve better performance compared to REVERB.

Distributed representations of words in a vector space have achieved considerable success in a wide range of NLP tasks [29,30], including applications to automatic speech recognition and machine translation [31, 32].

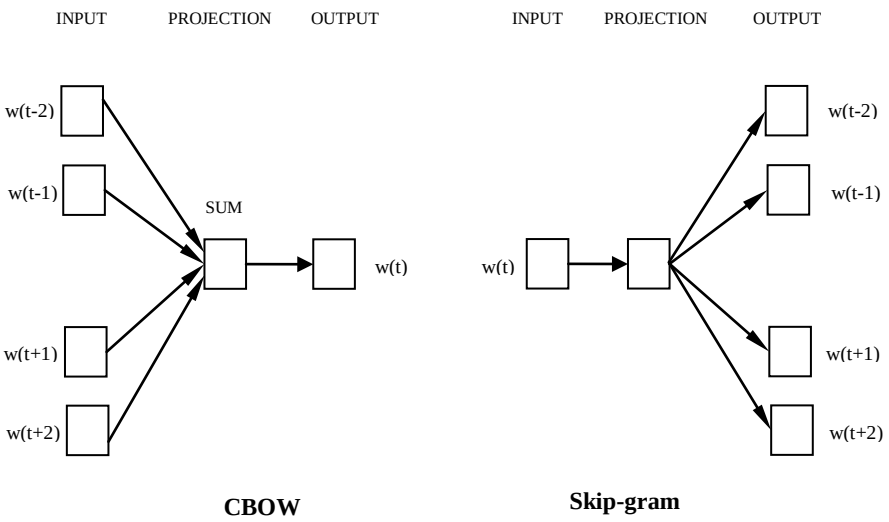


Fig. 2. the CBOW architecture predicts the current word based on the context and the skip-gram predicts the surrounding words given the current word.

Recently, Mikolov et al. introduced two novel algorithms for computing word vectors based on large unlabeled datasets. The first architecture is the continuous bag-of-words model (CBOW), while the second one is named as the Skip-gram model. Both the architectures are shown at **Fig. 2.** Given the surrounding words, the CBOW model predicts the current word, and the Skip-gram model predicts the surrounding words based on the current word.

The word representations computed with neural networks are very interesting because the word vectors learned from the model can explicitly capture many linguistic regularities and patterns. Surprisingly, many of these patterns can be represented as linear translations. For example, the vector calculation result of **"***Madrid"* - *"Spain"* + *"France"* is closer to *"Paris"* than to any other word vector [33, 34]. In other words, if two words are close to each other in linguist, they are "close" in vector space in terms of cosine distance. Inspired by this, we estimate whether a candidate is a correct relation according to the distance among the triple.

3. The Distance Approach

In this section, the Distance Approach is first introduced in detail, and in section 3.2 we prove that the new algorithm for ordering the candidate extractions is equivalent to Maximum Likelihood Estimation of the joint probability distribution over the three elements in the triple.

3.1 Algorithm of the Distance Approach

Open IE systems make a single or constant number of pass(es) over a corpus and extract a large number of relational triples $(Arg1, Pred, Arg2)$ without requiring any relation specific training data.

In our approach, we first extract candidate triples in two steps, which is proposed by REVERB as described in section 2. The candidate extraction is denoted as (X, R, Y), in which X and Y are NPs and R is a VP. REVERB uses a logistic regression classifier to evaluate the triples. In order to improve the performance of the extractor, we use the distance among the elements of a triple as the confidence score of the candidate extraction.

Assume S is a sequence of words $w_1, w_2, \dots, w_i, \dots, w_s$, and V_i is the word vector of w_i. According to the linear translations of word vectors, the 'word' vector for S can be calculated as:

$$V_S = \sum_{i=1}^{|S|} V_i \tag{1}$$

Where $|S|$ is the length of S. And for an unknown word, its word vector is zero.

Given a candidate extraction (X, R, Y), we could get the vector value of each element respectively

$$V_X = \sum_{i=1}^{|X|} V_i, \qquad V_R = \sum_{i=1}^{|R|} V_i, \qquad V_Y = \sum_{i=1}^{|Y|} V_i, \tag{2}$$

The distance (or similarity) among the three elements is computed separately, as follows:

$$d_{XR} = \frac{V_X^T V_R}{|V_X||V_R|}, \quad d_{RY} = \frac{V_R^T V_Y}{|V_R||V_Y|}, \quad d_{XY} = \frac{V_X^T V_Y}{|V_X||V_Y|} \tag{3}$$

where d_{XR} is the distance between $Arg1$ and $Pred$ in vector space, d_{RY} is the distance between $Pred$ and $Arg2$, and d_{XY} is the distance between $Arg1$ and $Arg2$. In a meaningful relational triple, there should be an appropriate distance between any two elements. The weighted average for the three distance values is:

$$P = n_1 d_{XR} + n_2 d_{RY} + n_3 d_{XY} \tag{4}$$

where n_1, n_2 and n_3 are the weights of the distance values, with $n_1 + n_2 + n_3 = 1$.

Because there is no clear evidence to determine the importance of the three distances, it could be supposed that $n_1 = n_2 = n_3 = 1/3$. After normalization, we can get $n_1 = n_2 = n_3 = 1$. Then the confidence score of the candidate extraction is represented as:

$$P = d_{XR} + d_{RY} + d_{XY} \tag{5}$$

where P is the average of the three distances.
Finally, in order to compare with the logistic classifier used in REVERB, the value of P is normalized to $\widetilde{P}$

$$\widetilde{P} = \frac{P}{\max\{P \text{ from all candidate extraction}\}} \tag{6}$$

with $0 \leq \widetilde{P} \leq 1$.

The distance-based algorithm is summarized in **Table 1**. The inputs of this algorithm include a set of word vectors and a collection of sentences, out of which candidate triples will be extracted. The outputs of the algorithm are a set of candidate extractions, and their corresponding confidence scores, based on the distance (or similarity) among the three elements.

Table 1. Algorithm of the distance approach

Input:	$L = \{S \mid S = w_1 w_2 \ldots w_m\}$. A collection of sentences. A sentence is a sequence of words $D = \{< w_i, v_i > \mid v_i \in R^n\}$. A dictionary constructed from an unlabeled dataset. In each entry, v_i is the word vector corresponding to word w_i, and v_i is based on word2vec from Google.
	1. For the set L, construct a collection of candidate extractions with the form (X, R, Y), where R is a relation phrase in a sentence within L, and R must satisfy the constraints shown in **Fig.1**. X is the nearest noun phrase on the left side of R, and Y is the nearest noun phrase on the right side of R. 2. For each candidate extraction (X, R, Y), calculate the 'vector' values for X, R and Y respectively, $V_X = \sum_{i=1}^{\lvert X\rvert} V_i, \quad V_R = \sum_{i=1}^{\lvert R\rvert} V_i, \quad V_Y = \sum_{i=1}^{\lvert Y\rvert} V_i,$ 3. Calculate the distance among the three elements X, R and Y separately, $d_{XR} = \frac{V_X^T V_R}{\lvert V_X\rvert\lvert V_R\rvert}, \quad d_{RY} = \frac{V_R^T V_Y}{\lvert V_R\rvert\lvert V_Y\rvert}, \quad d_{XY} = \frac{V_X^T V_Y}{\lvert V_X\rvert\lvert V_Y\rvert}$ 4. Calculate the confidence score for the candidate extraction (X, R, Y), $P = d_{XR} + d_{RY} + d_{XY}$ 5. Find the maximum value for all candidate extractions from L. $P_{max} = max\{P\ from\ all\ candidate\ extractions\}$ 6. For each candidate extraction, calculate its normalized confidence score. $\widetilde{P} = P/P_{max}$
Output:	$O = \{< (X, R, Y), \ \widetilde{P} >\}$. A set of candidate extractions along with their corresponding confidence scores, $\widetilde{P}$s.

For example, here is a candidate extraction {*he*, *is from*, *China*}. Suppose $vec(\text{"}he\text{"}) = [0.5, 0.3]$,$vec(\text{"}is\text{"}) = [0.1, 0.2]$, $vec(\text{"from"}) = [0.2, 0.4]$, $vec(\text{"}China\text{"}) = [0.6, 0.7]$, then $vec(\text{"}X\text{"}) = [0.5, 0.3]$, $vec(\text{"}R\text{"}) = vec(\text{"}is\text{"}) + vec(\text{"}from\text{"}) = [0.3, 0.6]$, $vec(\text{"}Y\text{"}) = [0.6, 0.7]$. Hence, we get $d_{XR} = 0.84$, $d_{RY} = 0.97$, $d_{XY} = 0.95$, and $P = d_{XR} + d_{RY} + d_{XY} = 2.76$. Therefore, if $max\{P\} = 2.90$, then $\widetilde{P} = 2.76/2.90 = 0.95$.

Next we will show that under certain conditions, P is equivalent to Maximum Likelihood Estimation of the joint probability distribution over the three elements X, R and Y, i.e. $P \propto max\, P_r(X,R,Y)$.

3.2 Proof for the algorithm

Lemma 1: Minimizing the cosine distance between two normalized vectors is equivalent to Maximizing the squared error between them.

Proof

Suppose $x, y \in R^n$, the angle between x and y is θ, with $\|x\| = 1$ and $\|y\| = 1$, then the squared error between x and y can be written

$$\begin{aligned} E(x,y) &= (x-y)^2 \\ &= \sum_{i=1}^{n} (x_i - y_i)^2 \\ &= x^2 + y^2 - 2\|x\|\|y\|\cos\theta \\ &= 2 - 2\cos\theta \end{aligned} \tag{7}$$

Where, $\cos\theta = \dfrac{x^T y}{\|x\|\|y\|}$

So, $E = 2 - 2cos\theta$. Since maximizing $cos\theta$ will minimize $E(x,y)$, thus

$$Min \sum_{i=1}^{n} (x_i - y_i)^2 = Max \cos\theta \tag{8}$$

This complete the proof.

□

Before we continue, we prove the following important lemma.

Lemma 2: Any learning algorithm whose loss function is defined as squared error will output a maximum likelihood hypothesis.

Proof

This theorem is applied to the problem of learning a continuous-valued target function, such as neural network learning, linear regression and polynomial curve fitting. For instance, the loss function of neural network is defined as squared error, and the network seeks the least-squared error hypothesis with gradient descent methods.

Suppose learner L works on an instance space X and hypothesis space H, where $X \subset R^n$, $H = \{h | h: X \to R\}$. The learner L is to learn an unknown target function $f: X \to R$ drawn from H.

suppose the training data set is $D = \{(x_1, d_1), (x_2, d_2), \ldots\ldots, (x_m, d_m)\}$, where the target value of each example is corrupted by random noise according to a Normal probability distribution. Formally, each training example (x_i, d_i) can be written as:

$$d_i = f(x_i) + e_i \qquad i = 1,2,\ldots,m \tag{9}$$

Where $f(x_i)$ is the noise-free value of the target function and e_i is a random variable representing the noise. According to the Center Limit Theorem, if there is a sufficiently large number of training examples, e_i obeys a Normal distribution. Next we will derive the maximum likelihood hypothesis

$$h_{ML} = \arg\max_{h \in H} p_r(D \mid h) \tag{10}$$

We assume a fixed set of training instances $\{x_1, x_2, \ldots\ldots, x_m\}$ and therefore consider the data D to be corresponding sequence of target values $D = \{d_1, d_2, \ldots\ldots, d_m\}$. Assuming the training examples are mutually independent given h, $p_r(D|h)$ can be written as the product of the various $p_r(d_i|h)$

$$h_{ML} = \arg\max_{h \in H} \prod_{i=1}^{m} p_r(d_i \mid h) \tag{11}$$

Given that the noise e_i obeys a Normal distribution with zero mean and unknown variance σ^2, each d_i must obey a Normal distribution with variance σ^2 centered around the true target value $f(x_i)$. Therefore $p_r(d_i|h)$ can be written as a Normal distribution with variance σ^2 and mean $\mu = f(x_i)$. Because we are writing the expression for the probability of d_i given that h is the correct description of the target function of f, we will substitute $\mu = f(x_i) = h(x_i)$, yielding

$$\begin{aligned} h_{ML} &= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-\mu)^2} \\ &= \arg\max_{h \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(d_i-h(x_i))^2} \end{aligned} \tag{12}$$

Because lnp_r is a monotonic function of p_r. Therefore maximizing lnp_r also maximizing p_r.

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}(d_i - h(x_i))^2 \tag{13}$$

The first term in this expression can be discarded because it is constant independent of h, yielding

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} -\frac{1}{2\sigma^2}(d_i - h(x_i))^2 \tag{14}$$

Maximizing this negative quantity is equivalent to minimizing the corresponding positive quantity,

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} \frac{1}{2\sigma^2}(d_i - h(x_i))^2 \tag{15}$$

Finally , we can again discard constants that are independent of h,

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2 \tag{16}$$

Thus, if h is the correct assumption, minimizing the squared error between d_i and $h(x_i)$ is equivalent to maximizing the joint probability distribution over the training examples. If we consider alternative independent assumption on various d_i , the equation $h_{ML} = argmax \prod p_r(d_i|h)$ will change to different form. For instance, in the theorem following, assuming that w_{t+j} is conditionally independent of w_t given h, and we can get $h_{ML} = argmax \prod p_r(w_{t+j}|w_t, h)$.

Next, another useful lemma is presented as following.

Lemma 3: In neural network learning, squared error cost function is equivalent to cross-entropy cost function.

Proof

Suppose the training data set is $D = \{(x^{(1)}, y_1), (x^{(2)}, y_2), \ldots\ldots, (x^{(m)}, y_m)\}$, with $x^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_i^{(j)}, \ldots, x_n^{(j)})^T \in R^n$, $y_j \in \{0, 1\}$. For a single neuron, if the weight vector is $W = (w_1, w_2, \ldots, w_i, \ldots, w_n)^T \in R^n$, the input value for the unit is $z^{(j)} = W^T x^{(j)}$. suppose the active function is sigmoid function or softmax function, as

$$a(z^{(j)}) = \frac{1}{1 + e^{-z^{(j)}}} \tag{17.a}$$

$$a(z^{(j)}) = \frac{e^{z^{(j)}}}{\sum_{i=1}^{m} e^{z^{(i)}}} \tag{17.b}$$

It holds that

$$a' = a(1 - a) \tag{18}$$

The squared cost function is defined as

$$J_1(W) = \frac{1}{2m} \sum_{j=1}^{m} (y_j - a(z^{(j)}))^2 \tag{19}$$

And the cross-entropy cost function is defined as

$$J_2(W) = -\frac{1}{m} \sum_{j=1}^{m} (y_j \ln a(z^{(j)}) + (1 - y_j) \ln(1 - a(z^{(j)}))) \tag{20}$$

For the squared error cost function, the gradient for weight vector W can be obtained by differentiating $J_1(W)$ from equation (19), as

$$\begin{aligned} \frac{\partial J_1(W)}{\partial w_i} &= \frac{\partial}{\partial w_i} \left(\frac{1}{2m} \sum_{j=1}^{m} (y_j - a(z^{(j)}))^2\right) \\ &= \frac{1}{m} \sum_{j=1}^{m} (a(z^{(j)}) - y_j) a'(z^{(j)}) x_i^{(j)} \end{aligned} \tag{21}$$

Because of the factor $a'(z^{(j)})$, the limitation of the equation (21) is that the gradient decreases with the increase of the error between target value and output value, and therefore slows down the convergency. To overcome this problem, we can discard this factor, $a'(z^{(j)})$, yielding

$$\frac{\partial J_1(W)}{\partial w_i} = \frac{1}{m}\sum_{j=1}^{m}(a(z^{(j)}) - y_j)x_i^{(j)} \tag{22}$$

In fact, for any cost function $J(W)$, we can calculate the gradient for weight vector W as

$$\begin{aligned}\frac{\partial J(W)}{\partial w_i} &= \frac{\partial J(W)}{\partial a(z^{(j)})}\cdot\frac{\partial a(z^{(j)})}{\partial z^{(j)}}\cdot\frac{\partial z^{(j)}}{\partial w_i} \\ &= \frac{\partial J(W)}{\partial a(z^{(j)})}\cdot a'(z^{(j)})\cdot\frac{\partial (W^T x^{(j)})}{\partial w_i} \\ &= \frac{\partial J(W)}{\partial a(z^{(j)})}\cdot a'(z^{(j)})\cdot x_i^{(j)} \\ &= \frac{\partial J(W)}{\partial a(z^{(j)})}\cdot a(z^{(j)})(1-a(z^{(j)}))\cdot x_i^{(j)}\end{aligned} \tag{23}$$

Comparing the equation (22) and equation (23), we obtain

$$\frac{\partial J(W)}{\partial a(z^{(j)})}\cdot a(z^{(j)})(1-a(z^{(j)}))\cdot x_i^{(j)} = \frac{1}{m}\sum_{j=1}^{m}(a(z^{(j)}) - y_j)x_i^{(j)} \tag{24}$$

or

$$dJ(W) = \frac{1}{m}\sum_{j=1}^{m}\frac{a(z^{(j)}) - y_j}{a(z^{(j)})(1-a(z^{(j)})}da(z^{(j)}) \tag{25}$$

Evalulating the integrals of equation (25), we have

$$\int dJ(W) = \int\frac{1}{m}\sum_{j=1}^{m}\frac{a(z^{(j)}) - y_j}{a(z^{(j)})(1-a(z^{(j)})}da(z^{(j)}) \tag{26}$$

yielding

$$J(W) = -\frac{1}{m}\sum_{j=1}^{m}(y_j \ln a(z^{(j)}) + (1-y_j)\ln(1-a(z^{(j)}))) + C \tag{27}$$

Where C is constant. Ignoring the constant C, we obtain

$$J(W) = -\frac{1}{m}\sum_{j=1}^{m}(y_j \ln a(z^{(j)}) + (1-y_j)\ln(1-a(z^{(j)}))) \tag{28}$$

It's the same as equation (20), so $J_1(W)$ is equivalent to $J_2(W)$.

On the other hand, assuming the cross-entropy cost function is utilized, the gradient for weight vector W can be obtained again by differentiating $J_2(W)$ from equation (20), as

$$\begin{aligned}\frac{\partial J_2(W)}{\partial w_i} &= \frac{\partial}{\partial w_i}(-\frac{1}{m}\sum_{j=1}^{m}(y_j \ln a(z^{(j)}) + (1-y_j)\ln(1-a(z^{(j)})))) \\ &= -\frac{1}{m}\sum_{j=1}^{m}\left(\frac{y_j}{a(z^{(j)})} - \frac{1-y_j}{1-a(z^{(j)})}\right)a'(z^{(j)})x_i^{(j)} \\ &= \frac{1}{m}\sum_{j=1}^{m}\frac{a'(z^{(j)})x_i^{(j)}}{a(z^{(j)})\left(1-a(z^{(j)})\right)}(a(z^{(j)})-y_j) \\ &= \frac{1}{m}\sum_{j=1}^{m}(a(z^{(j)})-y_j)x_i^{(j)} \end{aligned} \tag{29}$$

Notice this is the same as Equation (21) except for the extra term $a'(z^{(j)})$, which influences only convergency speed in neural network training.

So, it can be summarized that in neural network learning, squared error cost function is equivalent to cross-entropy cost function. And according to lemma 2, it can be concluded that any learning algorithm with cross-entropy cost function will also output a maximum likelihood hypothesis.

Now we are ready for our main result as following theorem.

Theorem: Assume (X, R, Y) is a candidate extraction. If Google word vector is employed, then $P \propto max\, P_r(X, R, Y)$, saying that P is equivalent to Maximum Likelihood Estimation of the joint probability distribution over the three elements X, R and Y. Here $P = d_{XR} + d_{RY} + d_{XY}$, as expressed in equation (5).

Proof

Word2vec is a typical neural network language model. In this model, the word vectors are initialized to Huffman code, and the loss function is defined as cross-entropy cost between the word vectors. As discussed in lemma 2 and lemma 3, when the network converged, it will output a maximum likelihood hypothesis. On the other hand, word2vec predicts the surrounding words with the current word or predicts the current word using the surrounding words. Given a sequence of words $D = w_1 w_2 \ldots w_T$, assuming the conditional independency, we can derive

$$\begin{aligned} p_r(D) &= p_r(w_1 w_2 \ldots w_T) \\ &= \prod_{t=1}^{T} p_r(w_t \mid w_{t+j}) \\ &= p_r(w_1 \mid w_2 \ldots w_{1+c})\, p_r(w_2 \mid w_1 w_3 \ldots w_{2+c}) \ldots p_r(w_t \mid w_{t-c} \ldots w_{t-1} w_{t+1} \ldots w_{t+c}) \ldots \end{aligned} \tag{30}$$

Where $-c \le j \le c, j \ne 0$

Assuming conditional independency on various w_{t+j}, it follows that

$$\begin{aligned}\ln p_r(D \mid h) &= \sum_{t=1}^{T} \ln p_r(w_{t+j} \mid w_t, h) \\ &= \sum_{t=1}^{T} \ln \prod_j p_r(w_{t+j} \mid w_t, h) \\ &= \sum_{t=1}^{T} \sum_j \ln p_r(w_{t+j} \mid w_t, h)\end{aligned} \tag{31}$$

As described in [25].

Assuming $p_r(w_{t+j} \mid w_t, h)$ converges to $\hat{p}_r(w_{t+j} \mid w_t)$, according to lemma 2, it can be written

$$h_{ML}(t) = \hat{p}_r(w_{t+j} \mid w_t) = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2 \tag{32}$$

By analyzing the network construction and search procedure of word2vec, it is clear that, in equation (32), x_i corresponds to w_{t+j} and d_i corresponds to w_t. When the network gets convergence, all w_{t+j} and w_t are projected to word vectors, $d_i = vec(w_t) = (x_1, x_2, \ldots\ldots, x_n)$, $h(x_i) = vec(w_{t+j}) = (y_1, y_2, \ldots\ldots, y_n)$. Thus,

$$\begin{aligned}h_{ML}(t) &= \hat{p}_r(w_{t+j} \mid w_t) \\ &= (vec(w_t) - vec(w_{t+j}))^2 \\ &= ((x_1, x_2, \ldots, x_n) - (y_1, y_2, \ldots, y_n))^2 \\ &= \sum_{i=1}^{n} (x_i - y_i)^2\end{aligned} \tag{33}$$

From equation (8) in lemma 1, equation (33) can be written as

$$\hat{p}_r(w_{t+j} \mid w_t) = \cos\theta = \frac{x^T y}{\|x\| \|y\|} \tag{34}$$

Suppose there is a candidate extraction (X, R, Y). The maximum likelihood estimation of $P_r(X, R, Y \mid h)$ is

$$\hat{p}_r(X, R, Y \mid h) = \hat{p}_r(X \mid R, Y, h)\, \hat{p}_r(R \mid Y, h)\, \hat{p}_r(Y \mid h) \tag{35}$$

Assuming X is conditionally independent of Y given h, we obtain

$$\hat{p}_r(X, R, Y \mid h) = \hat{p}_r(X \mid R, h)\, \hat{p}_r(R \mid Y, h)\, \hat{p}_r(Y \mid X, h) \tag{36}$$

According to equation (34), the equation (36) can be written as

$$\hat{p}_r(X,R,Y \mid h) = \frac{V_X^T V_R}{\|V_R\|} \cdot \frac{V_R^T V_Y}{\|V_R\|\|V_Y\|} \cdot \frac{V_X^T V_Y}{\|V_X\|\|V_Y\|} = d_{XR} \cdot d_{RY} \cdot d_{XY} \tag{37}$$

By taking logarithm on the above equation, we obtain

$$\ln \hat{p}_r(X,R,Y \mid h) = \ln d_{XR} + \ln d_{RY} + \ln d_{XY} \tag{38}$$

Finally, we have

$$\hat{p}_r(X,R,Y \mid h) \propto P = d_{XR} + d_{RY} + d_{XY} \tag{39}$$

By now, the proof for the proposed method is completed.

From the analyses above, it is concluded that:

(i) The three components in equation (5) represent the maximum likelihood estimation $\hat{p}_r(X \mid R,h)$、 $\hat{p}_r(R \mid Y,h)$ and $\hat{p}_r(Y \mid X,h)$ respectively

(ii) An alternative solution for equation (6) is Euclidean distance

$$P = \sqrt{(X-R)^2} + \sqrt{(R-Y)^2} + \sqrt{(Y-X)^2} \tag{40}$$

But the main disadvantage with Euclidean distance is that it is especially sensitive to the irrelevant attributes to determine the classification. And the cosine distance between two normalized vectors can avoid this problem as far as possible.

(iii) Another alternative solution is predicting R with X and Y, that is

$$P = p_r(R \mid X,Y) = \hat{p}_r(X,Y \mid R) \tag{41}$$

Supposing X is conditionally independent of Y given R, we have

$$P = \hat{p}_r(X \mid R)\hat{p}_r(Y \mid R) \tag{42}$$

taking logarithm on the above equation, then

$$\ln P = \ln \hat{p}_r(X \mid R) + \ln \hat{p}_r(Y \mid R) \propto \hat{p}_r(X \mid R) + \hat{p}_r(Y \mid R) = d_{XR} + d_{RY} \tag{43}$$

It is an equivalence of equation (5)

(iv) Google's word vector is the most excellent word vector by now, especially it's linear translations have achieved great attention among researchers. For example, the result of a vector calculation *"Madrid" - "Spain" + "France"* is closer to *"Paris"* than to any other word vector. That is why we calculate the value of a sequence as the sum of the word vectors within it. Nevertheless, there is no evidence that the linear translations are always correct. So, equation (1) may cause calculation error under certain condition. We should improve equation (1) in future work.

4. Experimental Results and Analysis

One of the challenges for Open IE systems is the lack of comparable datasets by now. Thus, researchers often construct test sets in their experiments independently. In our evaluation, the first test set is created from ACE2005, which is made up of heterogeneous texts. Since the main difference between REVERB and our algorithm is that they use different methods to assign confidence scores to candidates, we compare the two systems firstly on ACE2005 and determine the distance value at which our method achieves its best performance.

Because REVERB is regarded as a baseline system in Open IE, the REVERB test set (a combined set of WEB-500 and NYT-500) is considered as the benchmark dataset by the successive systems. We next compare our system against the newly proposed systems on this dataset. Since NYT-500 contains more complex sentences, and all of PCEOE, ArgOE, OpenIE4 can handle sentences with complex structures, we'll pay additional attention to the performance of these systems on NYT-500.

Only comparing methods based on effectiveness or efficiency can be misleading. Efficiency comparison is also performed in this experiment.

4.1 The experimental datasets and metrics

(1) The training dataset

As described in **Table 1**, we'll construct a dictionary of word vectors from an unlabeled dataset. The quality of the word vectors increases significantly with amount of the training data. In this research, the online dataset, latest Wikipedia dump (about 4.8GB), is used as training corpus. After pre-processing, we obtain more than 3 billion words in the clean text, including all kinds of inflections of the words, proper nouns, compound word, and combination words and so on. Then the Google word2vec was applied on the training set, and a collection of 300-dimensional word vectors were obtained.

(2) The test dadasets

We compare the distance approach against the recently proposed systems on three datasets. The first dataset is made up of 26 random English articles from the ACE2005 English corpus, including 5 Newswire articles, 5 Broadcast News articles, 4 Broadcast Conversation articles, 4 eblog articles, 4 Usenet Newsgroups articles and 4 conversational Telephone Speech articles. The second and the third datasets are Web-500 (consisting of 500 random Web sentences) and NYT-500 (consisting of 500 random sentences from the New York Times). These datasets vary in complexity. We evaluated the appearance of the complex sentences in each dataset. The high percentage of complex sentences strongly motivates the performance of dependency parsing procedures. **Table 2** shows the percentage of sentences containing no verb phrase, containing only one verb phrase, and containing more than one verb phrases. The percentage of semicolon and comma marks appearing in each dataset is also shown in **Table 2**.

Table 2. Complexity of the datasets

Dataset	Percentage of sentence contains VP=1	Percentage of sentence contains VP>1	Percentage of sentence contains semiclon	Percentage of sentence contains comma
ACE2005	25.3%	69.6%	6.5%	72.1%
NYT-500	19.2%	74.4%	8.4%	89.2%
WEB-500	41.6%	47.8%	2%	49%

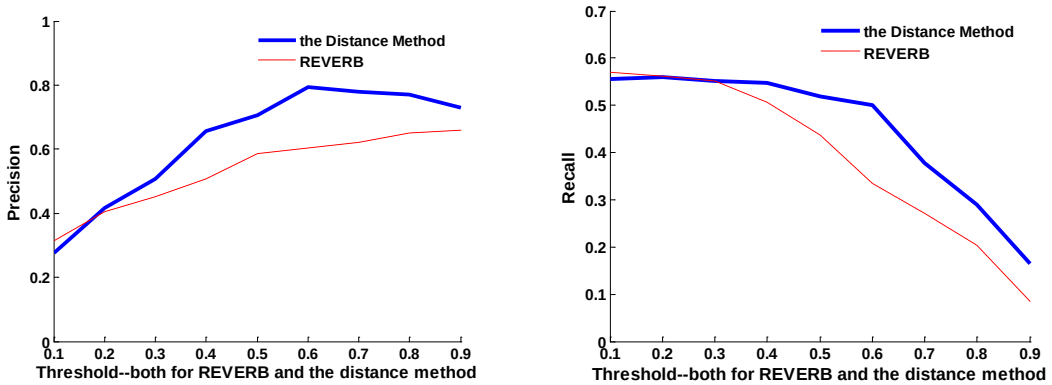
As is apparent from **Table 2**, the contents of ACE2005 and NYT-500 are dominated with complex sentences (sentences containing more than one verb phrase). The percentage of sentences with comma is also high in the ACE2005 and NYT-500 datasets. Based on the examination, it can be predicted that the dependency parsing procedures of PCEOE, ArgOE and OpenIE4 will have more effect on ACE2005 and NYT-500 rather than on WEB-500.

(3) The metrics used in this analysis

After triples were extracted from the datasets by each system, three judges with linguistic backgrounds evaluated the outputs of the systems and labeled whether the relation phrases were correct. The performance metrics used in this analysis are precision, recall and F-measure, defined as usual. Precision is the fraction of returned extractions that are correct. Recall is the fraction of correct extractions in the corpus that are returned. We use the total number of extractions from all systems labeled as correct by the judges as our measure of recall for the corpus.

4.2 Results of Experiments

This section compares the effectiveness and efficiency of the five methods, the distance method, REVERB, PCEOE, ArgOE and OpenIE4. In the first experiment, the confidence scores are used as thresholds in comparing our approach against REVERB on ACE2005. From this dataset, we got 1673 candidate extractions. As mentioned earlier, after the triples are extracted from the sentences, REVERB uses a logistic classifier's weight to order the triples, and the proposed approach evaluates the candidate extraction based on cosine distance among the elements. The logistic classifier is from Weka's implementation. REVERB trains the classifier on a hand-labeled development set of random Web sentences, and the selected features just include shallow syntactic features and POS. In this paper, we use Google's word vector to calculate the distance value. The word vectors are trained on the latest Wikipedia dump. **Fig. 3(a)** illustrates precision-threshold (confidence) curves for the two methods. **Fig. 3(b)** presents recall-threshold (confidence) curves. Note here that the thresholds for the distance method are distance values (normalized), and the thresholds for REVERB are the confidence scores based on logistic classifier.



(a) Precision-confidence (threshold) **(b)** Recall-confidence (threshold)

Fig. 3. Performance comparison on ACE2005 dataset, using confidence scores as thresholds. REVERB assigns confidence scores to the candidates with a logistic classifier, while the proposed method calculates confidence scores based on distance values in vector space.

From the figures above, it is obvious that when thresholds are higher than 0.2, the distance method gets higher precision than REVERB. And when thresholds are higher than 0.3, the distance method gets higher recall again. **Fig. 4** shows F1 scores of the two methods. The distance method gets its peak value 0.61 at the threshold 0.6. Since F1 scores represent the average of precision and recall, we can conclude that the distance method makes some improvements over REVERB.

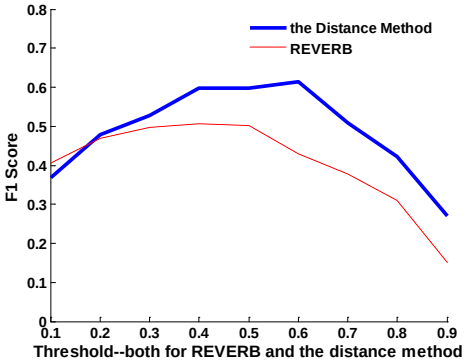
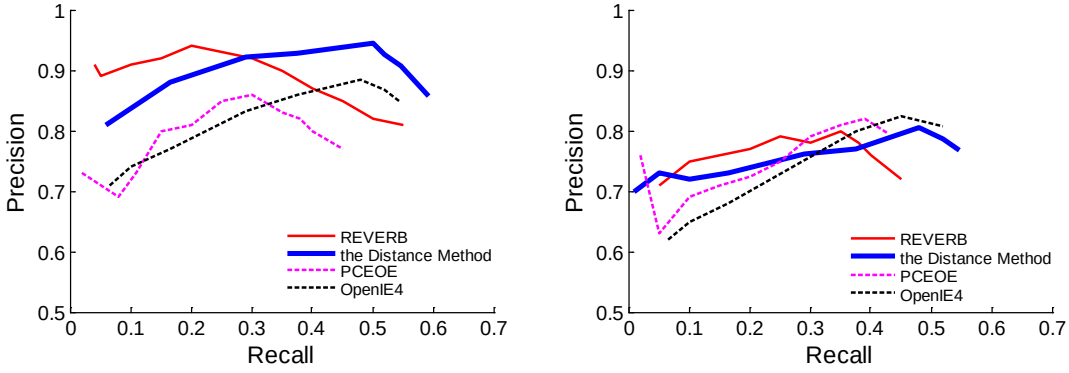


Fig. 4. F1-confidence (threshold)

The next dataset consists of NYT-500 and WEB-500, with 1000 sentences in total. This is the same test set used in [1] to evaluate REVERB system, and was reused in ArgOE**.** Because the sentences in this dataset are much simpler than those in ACE2005, both REVERB and the proposed method receive a boost to precision on this dataset. Each system (except for ArgOE) returns confidence scores for its extractions. For a given threshold, we can measure the precision and recall of the output. **Fig. 5** presents the precision-recall curves for the different Open IE systems.



(a) Performance on WEB-500&NYT-500 **(b)** Performance on NYT-500 only

Fig. 5. Precision-Recall curves for the different Open IE systems

As **Fig. 5(a)** shows, when recalls are higher than 0.3, the proposed method has higher precision than all the comparison systems. The first row of **Table 3** compares the highest F-measure of these systems on the combined set of NYT-500 and WEB-500. The proposed approach has over 30% higher F-measure than OpenIE4 and more than double the F-measure of REVERB, PCEOE and ArgOE.

Table 3. Effectiveness comparison of the systems

Methods / Datasets	REVERB			PCEOE			ArgOE			OpenIE4			Proposed		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
WEB-500&NYT-500	.92	.29	.44	.86	.31	.46	.60	.28	.26	.89	.50	.64	.95	.52	.67
NYT-500 only	.78	.36	.49	.82	.39	.53	.43	.16	.23	.83	.45	.58	.81	.48	.60

The lower recall of REVERB is due to its dependency on logistic classifier, which requires human-labeled corpus in its learning. Unfortunately, there is so far no sufficient training data suitable for open IE tasks. Thus, the performance of the logistic classifier is restricted. On the contrary, the proposed method is based on Google's word vector, which is an unsupervised learning approach and needs only unlabeled training corpus. Because unlabeled training data can be easily acquired on the Web, together with the much lower computational complexity of Google's algorithm, it is possible to compute very accurate high dimensional word vectors from a much larger dataset. That's why our distance approach outperforms REVERB. Similarly, OpenIE4 relies also on supervised learning to train its classifier and suffers from the lack of training data. A common problem for dependency-based Open IE systems (e.g. PCEOE and ArgOE) is the large influence of parser errors. One possible reason for the comparably poor performance of PCEOE and ArgOE might be the lower parsing performance. In summary, the proposed method clearly outperforms the other systems in this dataset, in terms of both precision and recall.

Since the other three systems, PCEOE, ArgOE and OpenIE4 can handle complex sentences, as reported, we compare the proposed approach with these systems on the more complex dataset, NYT-500, which is the same test set used in [21]. **Fig. 5(b)** illustrates recall-precision curves on this dataset. The highest precision and F-measure of these systems are shown in the second row of **Table 3.**

Since the filtering procedure has positive effect on precision, PCEOE gets slightly higher precision than our proposed approach. Nevertheless, the proposed approach presents 23% higher recall than the more sophisticated PCEOE, somewhat surprisingly. The reason is that PCEOE discards a large amount of sentences during its filtering procedure. OpenIE4 achieves slightly higher precision and lower recall than the proposed method, since its learning-based algorithm benefits precision and parsing errors decrease recall. The poor performance of ArgOE is again due to the parser errors. Overall, the F-measure value of our proposed approach in this dataset outperforms the previous work.

In order to scale to Web size corpora, Open IE systems are strongly in favor of speed. Thus, it is very important to compare the methods based on efficiency (computational cost). We have measured the number of output triples per second for each method in this experiment, without considering initialization or loading any libraries or models in memory. Performance comparisons of these systems in different datasets are depicted in **Table 4,** in terms of efficiency. To ensure a fair comparison, we make sure each method runs in a singlethreaded mode, thus utilizing a single computing core at all times.

Table 4. Efficiency comparison of the five methods (triples/second)

Methods / Datasets	REVERB	PCEOE	ArgOE	OpenIE4	Proposed
WEB-500	39.6	37.5	38.1	34.2	42.8
NYT-500	35.8	32.6	34.5	30.1	37.1
ACE2005	37.2	36.7	36.1	32.5	40.6

For efficiency, our proposed approach outperforms the other systems in all datasets. The proposed approach gets 13% increase over PCEOE, averaged over all datasets. This is because PCEOE employs dependency parsing in its clause extraction procedure, which is a time consuming step. Because of the similar reason, the proposed approach gets 11% increase over ArgOE and achieves 25% increase over OpenIE4 in efficiency. The proposed approach also achieves 7% increase over REVERB in efficiency, since REVERB relies on a logistic classifier to evaluate the candidate extractions and the classifier must perform feature extraction and some necessary computation. On the other hand, although the proposed approach need to search vectors for the words within candidate extractions to compute the confidence score, the word vectors are stored in hashtable, which ensures the high efficiency over REVERB.

5. Conclusion and Future Work

A great variety of Open IE systems have been developed in recent years. REVERB is the second generation open information extraction system, which focuses on identifying more meaningful and informative relation phrases. It outperforms the previous Open IE systems by significant margins and uses a logistic regression classifier to check the candidate extraction. This work presents a purely unsupervised learning algorithm for Open IE based on Google word vector. Different from REVERB, the proposed method uses the distance values among the extraction elements as confidence score of the candidate extraction. The new approach is proved to be an equivalence to Maximum Likelihood Estimation via Bayesian Analysis and Artificial Neural Network theory. The proveness itself also suggests theoretically a typical usage of word vector for other NLP tasks. We evaluated the new method on three benchmark datasets. Experiments show that the proposed method can lead to further improvements over the newly presented open IE systems, in terms of effectiveness and efficiency. The limitation of the proposed method is that this algorithm just calculates the arithmetic average of three distances. We should improve equation (1) in future work.

While much of the previous works on Open IE aimed at verb-mediated relations, some recent systems are focusing on Nominal Open IE [35], which extracts open relational tuples mediated by nouns (not verbs). This is an important subtask of Open IE, since many relations are frequently expressed via nouns, instead of verbs. We will do research in this aspect in future work.

References

[1] Oren Etzioni, Anthony Fader，“Open Information Extraction: the Second Generation,” in *Proc. of Int. Joint Conf. on Artificial Intelligence* , vol.8, no.4, pp.3-10, August 3-9, 2013. Article (CrossRef Link)

[2] Michele Banko, Michael J. Cafarella, “ Open information extraction from the web.” in *Proc. of Int. Joint Conf. on Artificial Intelligence* , vol.12, no.51, pp.68-74, January 6-12, 2007. Article(CrossRefLink)

[3] Michele Banko and Oren Etzioni, “The tradeoffs between open and traditional relation extraction,” in *Proc. of Annual Meeting of the Association for Computational Linguistics*,” pp.28-36 June 15-20, 2008. Article(CrossRefLink)

[4] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, “StatSnowball: a statistical approach to extracting entity relationships,” in *Proc. of Int. Conf. on World Wide Web*, pp.101-110, April 20-24, 2009. Article(CrossRefLink)

[5] Fei Wu, Daniel S. Weld, "Open information extraction using Wikipedia," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, pp.118-127, July 11-16, 2010. Article(CrossRefLink)
[6] M. Schmitz, R. Bart, S. Soderland, O. Etzioni, "Open language learning for information extraction," in *Proc. of Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 523–534 , July 12-14, 2012. Article(CrossRefLink)
[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell, "Toward an Architecture for Never-Ending Language Learning," in *Proc. of Conf. on Artificial Intelligence*, pp. 1306-1313, July 11–15, 2010. Article(CrossRefLink)
[8] A. Akbik and J. Broß, "Wanderlust: Extracting semantic relations from natural language text using dependency grammar patterns," in *Proc. of IEEE Int. Conf. on Data Mining* pp. 6-15, December 6-10, 2009. Article(CrossRefLink)
[9] F. M. Suchanek, M. Sozio, and G. Weikum, "*SOFIE:* a self-organizing framework for information extraction," in *Proc. of Int. Conf. on World Wide Web*, pp. 631–640, April 20-24, 2009. Article(CrossRefLink)
[10] N. Nakashole, M. Theobald, and G. Weikum, "Scalable knowledge harvesting with high precision and high recall," in *Proc. of ACM Int. Conf. on Web Search and Data Mining*, pp. 227–236, February 9-12, 2011. Article(CrossRefLink)
[11] N. Nakashole, G. Weikum, and F. Suchanek, "Discovering Semantic Relations from the Web and Organizing them with PATTY," *ACM SIGMOD Record*, vol. 42, no. 2, pp. 29–34, June, 2013. Article(CrossRefLink)
[12] F. Mesquita, "Clustering techniques for open relation extraction," in *Proc. of the SIGMOD/PODS 2012 PhD Symposium*, pp. 27–32, May 20th, 2012. Article(CrossRefLink)
[13] F. Mesquita, J. Schmidek, and D. Barbosa, "Effectiveness and Efficiency of Open Relation Extraction," in *Proc. of the 2013 Conf. on Empirical Methods in Natural Language Processing*, pp. 447–457, October 18–21, 2013. Article(CrossRefLink)
[14] M. Miwa, R. Saetre, Y. Miyao, and J. Tsujii, "Entity-focused sentence simplification for relation extraction," in *Proc. of the 23rd International Conf. on Computational Linguistics*, pp. 788–796, August 23-27, 2010. Article(CrossRefLink)
[15] I. Segura-Bedmar, P. Martnez and C. de Pablo-Sánchez, "A linguistic rule-based approach to extract drug-drug interactions from pharmacological documents," in *Proc. of the 4th Int. Workshop on Data and Text Mining in Biomedical Informatics*, vol. 12, no. Suppl 2, pp. 1-11, March 29, 2011. Article(CrossRefLink)
[16] J. Schmidek, D. Barbosa, "Improving Open Relation Extraction via Sentence Re-Structuring," in *Proc. of Int. Conf. on Language Resources and Evaluation*, pp.3720-3723, May 26-31, 2014. Article(CrossRefLink)
[17] G. Angeli, M. J. Premkumar, and C. D. Manning, "Leveraging Linguistic Structure For Open Domain Information Extraction," in *Proc. of Annual Meeting of the Association for Computational Linguistics*, pp.344-354, July 26-31, 2015. Article(CrossRefLink)
[18] L. Del Corro, R. Gemulla, "ClausIE: clause-based open information extraction," in *Proc. of the Int. Conf. on World Wide Web*, pp. 355–366, May 13–17, 2013. Article(CrossRefLink)
[19] R. Chandrasekar, C. Doran, and B. Srinivas, "Motivations and methods for text simplification," in *Proc. of the 16th Conf. on Computational Linguistics*,vol.2, pp. 1041–1044, August 5-9, 1996. Article(CrossRefLink)
[20] I. Dornescu, R. Evans, and C. Orasan, "Relative clause extraction for syntactic simplification," in *Proc. of the Workshop on Automatic Text Simplification-Methods and Applications in the Multilingual Society*, pp. 1–10, August 24th, 2014. Article(CrossRefLink)
[21] Ade Romadhony, Dwi H. Widyantoro, Ayu Purwariant, "Phrase-based Clause Extraction for Open Information Extraction System," in *Proc. of the 7th International Conf. on Advanced Computer Science and Information Systems*, pp.156-162, October 10-11, 2015. Article(CrossRefLink)
[22] Gamallo P, Garcia M, "Multilingual Open Information Extraction," in *Proc. of the 17th Portuguese Conf. on Artificial Inteligence*, pp.711-722, September 8-11, 2015.

Article(CrossRefLink)
[23] Janara Christensen , Mausam, "An analysis of open information extraction based on semantic role labeling," in *Proc. of International Conf. on Knowledge Capture*, vol.34, pp. 113-120, June 23-26, 2011. Article(CrossRefLink)
[24] Mausam, "Open Information Extraction Systems and Downstream Applications," in *Proc. of the 25th International Joint Conf. on Artificial Intelligence*, pp.4074-4077, July 9-15, 2016. Article(CrossRefLink)
[25] Tomas Mikolov, Kai Chen, "Efficient Estimation of Word Representations in Vector Space," in *Proc. of Workshop at International Conf. on Learning Representations*, pp. 65-76, May 2-4, 2013. Article(CrossRefLink)
[26] A. Siddharthan, "A survey of research on text simplification," *International Journal of Applied Linguistics*, vol. 165, no. 2, pp. 259–298, March, 2014. Article(CrossRefLink)
[27] A. Akbik and A. Löser, "Kraken: N-ary facts in open information extraction," in *Proc. of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pp. 52–56, June 7-8, 2012,. Article(CrossRefLink)
[28] AmalZouaq, MichelGagnon, "An assessment of open relation extraction systems for the semantic web," *Information Systems*, vol. 71, pp.228-239, November, 2017. Article(CrossRefLink)
[29] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: deep neural networks with multitask learning," in *Proc. of the 25th Int. Conf. on Machine Learning*, pp.160-167, July 5-9, 2008. Article(CrossRefLink)
[30] Peter D. Turney, "Distributional semantics beyond words: Supervised learning of analogy and paraphrase," *Transactions of the Association for Computational Linguistics*, vol. 1, pp. 353-366, February, 2013. Article(CrossRefLink)
[31] Holger Schwenk, "Continuous space language models," *Computer Speech and Language*, vol. 21, No.3, pp. 492-518, July, 2007. Article(CrossRefLink)
[32] Tomas Mikolov, "Statistical Language Models Based on Neural Networks," *PhD Thesis, Brno University of Technology*, 2012. Article(CrossRefLink)
[33] Tomas Mikolov, Kai Chen, "Efficient estimation of word representations in vector space," in *Proc. of Workshop at Int. Conf. on Learning Representations*, pp. 65-76, May 2-4, 2013. Article(CrossRefLink)
[34] Tomas Mikolov, Wen-tau Yih and Geoffrey Zweig, "Linguistic Regularities in Continuous Space Word Representations," in *Proc. of Conf. of the North American Chapter of the Association for Computational Linguistics*, pp. 746-751, June 9-14, 2013. Article(CrossRefLink)
[35] Harinder Pal, Mausam, "Demonyms and Compound Relational Nouns in Nominal Open IE," in *Proc. of the 5th Workshop on Automated Knowledge Base Construction*, pp 35-39, June 17th, 2016. Article(CrossRefLink)

Liu Peiqian received his M.S. degree from Taiyuan University of technology , Taiyuan, China. He is currently working toward the Ph.D. degree in the Department of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include networking, natural language processing, and machine learning.

Wang Xiaojie received his Ph.D. degree from Beihang University ,Beijing, China. He is currently a professor at Department of Computer Science, Beijing University of Posts and Telecommunications. His research interests include natural language processing and cognitive computing.