Differential Evolutionary Algorithm Based on Multiple Vector Metrics for Semantic Similarity Assessment in Continuous Vector Space

Yuanyuan Cai, Wei Lu, Xiaoping Che, Kailun Shi
School of Software Engineering
Beijing Jiaotong University
Beijing, China
Email: {yycai,luwei,xpche,shikailun}@bjtu.edu.cn

Abstract

Automatic service discovery in heterogeneous environment is becoming one of the challenging problems for applications in semantic web, wireless sensor networks, etc. It is mainly due to the lack of accurate semantic similarity assessment between profile attributes of user request and web services. Generally, lexical semantic resources consist of corpus and domain knowledge. To improve similarity measures in terms of accuracy, various hybrid methods have been proposed to either integrate different semantic resources or combine various similarity methods based on a single resource. In this work, we propose a novel approach which combines vector similarity metrics in a continuous vector space to evaluate semantic similarity between concepts. This approach takes advantage of both corpus and knowledge base by constructing diverse vector space models. Specifically, we use differential evolutionary (DE) algorithm which is an powerful population-based stochastic search strategy for obtaining optimal value of the combination. Our approach has been validated against a variety of vector-based similarity approaches on multiple benchmark datasets. The empirical results demonstrate that our approach outperforms the state-of-the-art approaches. The results also indicate the continuous vectors are efficient for evaluating semantic similarity, since they have outstanding expressiveness to latent semantic features of words. Moreover, the robustness of our approach is presented by the steady measure results under different hyper-parameters of neural network.

Keywords-differential evolutionary; semantic similarity; continuous vector space; vector similarity metrics

1 Introduction

The vast number of information and heterogeneous resources distributed on the web have made the semantic analysis and semantic interoperability more challenging, especially in some fields such as semantic web, natural language processing (NLP) and social network. Semantic similarity measurement for concepts, which measures the degree of similarity or dissimilarity between two concepts, enables the precise service discovery and information inquiry. For example, a user who is querying the *bank* service can obtain results consisting of the words *deposit* and *interests* rather than *slope* and *river*. Hence, the semantic similarity measurement for concepts has been an attractive research content and also an important component in the related applications, such as automated service discovery [27], text classification [15] and emotion mining [4].

Existing approaches to measuring semantic similarity between concepts can be divided into corpus-based and knowledge-based approaches in terms of the semantic resources available. Corpus-based approaches primarily map a given corpus into a vector space [37] to compute the similarity between lexicon vectors. The words close together in the vector space tend to be semantically similar or occur in similar contexts. In these approaches, semantic features of words derive from the distributional properties of words in statistic corpus, which consist of the distribution and the frequency of lexical context. Corpus-based approaches are limited to the distributional VSM based on lexical co-occurrence statistics in corpus, since the vectors are modeled by "bag of words" which scratch the surface of words without reflecting sufficient semantic association of words. To explicitly decode implied semantic information from corpus into the distributional vector space, some related works leverage dimension reduction technologies such as Latent Semantic Analysis (LSA) [12], Latent Dirichlet Allocation (LDA) [8] and distributional information simi-

DOI reference number: 10.18293/DMS2015-001

lairity [20]. However, these works still use discrete vectors which lack the powerful expression capability of latent semantic and syntactic information. Therefore, rare and polysemous words are often poorly estimated.

Knowledge-based approaches take advantage of preexisting knowledge bases such as thesauri and WordNet ontology [24] to measure semantic similarity. In terms of semantic properties used in semantic computations, WordNet-based measures can be roughly classified into path-based, information content (IC)-based, feature-based and hybrid measures. The path-based measures and the IC-based measures mainly exploit the path difference and IC difference between concepts, while the feature-based measures rely on constructing concept vectors based on intrinsic properties of concepts and computing the similarity between vectors. As the feature-based approaches, gloss overlaps [6] and the cosine similarity between gloss vectors [28] can be directly used to measure semantic similarity. Liu et al. took local densities as the intrinsic properties of concepts and computed the cosine similarity of concept vectors for measuring semantic similarity between concepts [21].

To capture different aspects of semantic similarity between concepts, a variety of combined strategies are proposed, in terms of different measures and heterogeneous semantic resources. Yih and Qazvinian incorporated different vector measurements based on the heterogeneous lexical sources such as Wikipedia, web search engine, thesaurus and WordNet [35]. Alves et al. proposed a regression function where lexical similarity, syntactic similarity, semantic similarity and distributional similarity are input as independent variables [2]. Similarly, Bär et al. introduced a linear regression model integrating multiple content similarity values at the aspects of string, semantic, structure, etc [7]. Chaves-González and MartíNez-Gil combined WordNet-based semantic similarity measures using a metaheuristic algorithm to find a optimized solution [9]. Mihalcea et al. focused on the corpus-based cosine similarity and WordNet-based similarity [5]. In their approach, the distributed word vectors were linearly aggregated into diverse level representation related to phrase, sentence and paragraph. These hybrid approaches integrate different vector space models or different similarity methods with a single resource. However, few measures focus on the combination of vector similarity metrics for semantic similarity measurement.

This work contributes to integrating various vector similarity metrics such as cosine distance and Euclidean distance using a differential evolutionary (DE) algorithm. We assume that different metrics can induce varying degrees of semantic similarity between concepts. E.g., the cosine distance determinates the angle distance between two vectors (directional similarity) in the vector space, whereas the Euclidean distance evaluates straight-line distance between two vectors (magnitude similarity). Hence, in this work, fine-grained semantic similarities from different aspects are provided by a variety of metrics to optimize the similarity measurement. We use a DE algorithm to combine different vector-based similarity measures which rely on either corpus or WordNet. Furthermore, inspired by the application of distributed word representation from deep learning [22], we measure semantic similarity in the continuous vector space which reveals latent semantics. In addition, we conduct an additional experimentation to study the effects of various similarity metrics and hyper-parameters of neural network on the results of semantic comparison, since some systematical investigations indicated that the vector-based similarity approaches highly depend on the quality of VSM construction.

The rest of this paper is organized as follows: the related works are presented in Section 2. The problem and similarity metrics we used in this work are summarized in Section 3. Our methodology and experimental results on several evaluation criterions are discussed in Section 4. Conclusions and future work are given in Section 5.

2 Related works

Previous semantic similarity measures take advantage of domain ontology or corpus to compute the similarity between words. Ontology-based measures focus on exploring structure properties of ontology in semantic similarity computation, while corpus-based measures are based on the similarity of discrete vectors and improved by the technologies of dimensionality reduction. As an alternative of discrete vector model, the continuous word representation derived from deep learning has significantly benefited the vector-based semantic similarity measurement recently [36]. Continuous word representation, namely distributed word embedding, is a real-valued vector whose each dimension represents a latent semantic feature of words. In the continuous VSM, the words are encoded within a low-dimension vectors via unsupervised neural network training, which can better understand the significance and syntactic structure of words in a corpus text. With the powerful expressiveness of latent semantics, the continuous VSMs contribute to the outstanding performances of semantic disambiguation and analogy reasoning as well as other tasks [18]. Specially, according to Mikolov [23], the continuous word representations are independent across languages in terms of analogy relationship of word pairs.

Similarity metric or distance metric is an important part of vector similarity measures. When evaluating the semantic similarity of concepts, most works perform with a single computational metric, such as vector overlaps, cosine distance and Euclidean distance [1]. Based on the cosine similarity of vectors, Faruqui and Dyer evaluated the concept

similarity and the diversity of continuous word embeddings derived from different natural networks [13]. Pennington et al. learned distributed vectors from unsupervised global log-bilinear regression model with matrix factorization, and took the cosine value of the vectors as concept similarity [29]. However, a single metric could not capture all the aspects of semantic similarity and suit all types of input data. In addition, some works focus on studying effects of various similarity metrics on semantic similarity measurement. These studies contribute to the integration of different computational metrics. As an instance, Kiela and Clark studied the computational metric, data source, dimensionality reduction strategy, term weighting scheme and the parameters of vectors including window size and feature granularity in similarity tasks [19]. However, their evaluation concentrated on the distributional vector models. As regards continuous distributed vector model, Hill et al. demonstrated that the larger training windows work better for measuring similarity of abstract words than concrete words, and vice versa [17]. Chen et al. found that the lower dimensions of word embeddings significantly drop the accuracy of the classifiers across all the publicly available word embeddings [10]. Inspired by these work, we focus on the continuous vector space. Differing from other studies on similarity measures, we take advantage of vector similarity metrics.

Instead of proposing a new vector similarity metric, our study aims to improve the evaluation results obtained in single metric by combining multiple vector similarity functions. Hence, we propose a combination strategy to assessing semantic similarity based on the differential evolutionary (DE) algorithm. The algorithm of DE [34] is a population-based stochastic search strategy for solving global optimization problems. It derives from evolutionary algorithm (EA) and has multiple variants according to the strategy for generation of new candidate members [11, 26]. These variants have been proved applicable for continuous function optimization in a large number of research domains such as heat transfer [3].

3 Semantic similarity measurement based on differential evolutionary algorithm

In this section, we define the problem and research object on similarity evaluation, and describe the proposed hybrid measure which incorporates the heterogeneous similarity metrics for vector via differential evolutionary algorithm. In this work, the differential evolution algorithm is used for addressing the problem of the incorporation of various metrics, since it offers competitive solutions for evaluating the different aspects of semantic similarity. It iteratively assigns each similarity metric a specific weight. Fig. 1 illustrates the DE algorithm in our work. It performs with the similarity values provided by various vector-based metrics. All the metrics evenly contribute to evaluate the degree of semantic similarity between two concepts at the beginning of the differential evolution. Then the metric which provides the most similar results to the human judgement is offered the highest weight after automatic evolution process consists of initialization, mutation, crossover and selection.

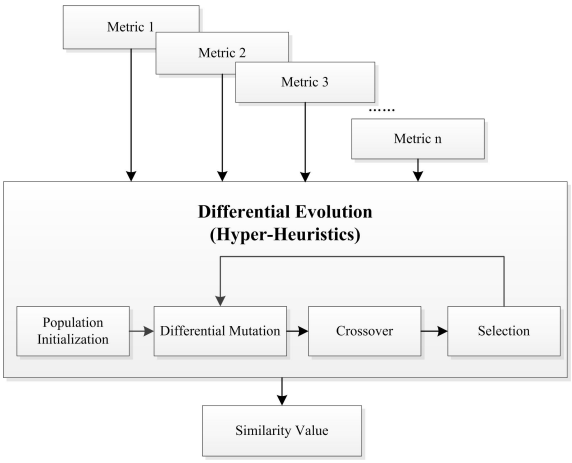

Figure 1. Illustrative workflow of the differential evolution (DE) algorithm.

3.1 Problem definition

There are given two concepts C_1 and C_2, the problem is to determine the degree of their semantic similarity. The vector-based semantic similarity calculation not only depends on the quality of vector but also involves the vector distance metric. Hence, we adopt various similarity metrics with low-dimensional continuous vectors. Each metric focuses on different lexical semantic relations between concepts consist of synonymy, hypernymy, hyponym and even antonymy, as well as co-occurrence relation [38], which respectively provide a certain degree of semantic similarity. Based on the combination strategy, we realize the integration of different metrics to capture semantic relations and determine semantic similarity between vectors. Formally, we define the two concepts as vector X and Y.

3.2 Vector similarity metrics

There exist numbers of metrics for vector similarity computation. Table 1 summarizes the similarity metrics explored in our work for two concept vectors. The first column indicates the general type of metrics and the second column gives their formalized definition. And the third column presents a brief explanation of the metrics.

From the perspective of vector direction, cosine metric measures how similar two vectors are. On the contrary, Eu-

Table 1. Similarity metrics between n-dimensional vector X and Y.

Similarity measure	Function definition	Description
Cosine	$\frac{X \cdot Y}{\|X\| \cdot \|Y\|}$	Cosine similarity computes cosine value of the vectorial angle in vector space
Euclidean	$\frac{1}{1+\|X-Y\|}$	Euclidean distance evaluates the absolute length of the line segment which connects the terminal points of two vector
Manhattan	$\frac{1}{1+\sum_{i=1}^{n}\|X_i-Y_i\|}$	Also known as the Cityblock distance, which is only possible to travel directly along pixel grid lines when going from one pixel to the other
Chebyshev	$\frac{1}{1+max_i\|X_i-Y_i\|}$	Chebyshev distance evaluates the maximum of the absolute distances in each dimension of vectors
Correlation	$\frac{(X-\overline{X})\cdot(Y-\overline{Y})}{\|X\| \cdot \|Y\|}$	Correlation distance evaluates the degree of linear correlation between vectors
Tanimoto	$\frac{X \cdot Y}{\|X\|+\|Y\|-X \cdot Y}$	Tanimoto similarity measures the degree of shared features between two vectors

clidean distance which is sensitive to the absolute difference of individual numerical features provides us the magnitude of the difference between two vectors. Other distance measures such as Manhattan distance and Chebyshev distance evaluate the sum or the maximum of differences on the features of vectors. Correlation distance contributes to revealing the linear association between two vectors. The Tanimoto coefficient is used to measure matching degree of the features between two vectors.

3.3 Differential evolution algorithm

The hyper-heuristics DE algorithm works as a solution for the global optimization of the combination of vector metrics. It holds a population with the size of NP and defines each member of the population as a candidate solution that a vector of weighting coefficients. In the evolution process, new individuals are generated due to the difference between the chosen individuals (see Fig. 2). Table 2 profiles the individuals in population, where each dimensionality of the individuals represents a similarity metric M_k whose similarity result weighted by and the coefficient $w(M_k)$.

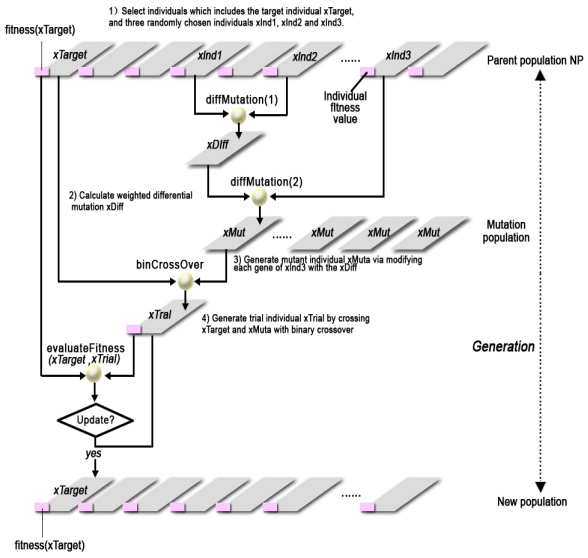

Figure 2. Profile of the rand/1/bin differential evolution (DE) algorithm.

Table 2. Individual profile.

Metric 1	Metric 2	Metric 3	...	Metric N
$w(M_1)$	$w(M_2)$	$w(M_3)$	...	$w(M_N)$

One individual in a population is represented as a vector like $\vec{I}$ = [$w(M_1)$,$w(M_2)$,. . .,$w(M_N)$] where each element $w(M_k) \in$ [MIN,MAX] is a real number. To some extent, the task of DE algorithm is a search for a vector $\vec{I^*}$ to optimize the objective function of the given problem. DE performs the evolution of NP individuals $\vec{I_{ik}}$ with N dimensions (i=1,2,. . .,NP; k=1,2,. . .,N) in a vast search space. It consists of three basic operations that mutation, crossover and selection. Among the existing variants of DE algorithm, we choose the strategy $rand/1/bin$ [34] in this work, in terms of the scheme of mutation and crossover as well as selection. The notation $rand/1/bin$ indicates how the mutation and crossover operators work. That is, the DE algorithm selects individuals at random, then adopts binomial crossover (bin) and a unique difference vector (/1/) to generate the mutation of the random individual (rand) in the parent population. Fig. 2 illustrates the $rand/1/bin$ strategy, and its configures are detailed in Section 4. This strategy

starts with the random generation of the population through assigning a random weights to each gene of the individual.

The main process of the DE algorithm initiates after calculating fitness for the whole population. DE algorithm selects the individuals consisting of target individual $\vec{I_t}$, and three randomly chosen individuals $\vec{I_{r1}}$, $\vec{I_{r2}}$, $\vec{I_{r3}}$. Then the weighted differential mutation $\delta\vec{I}$ is calculated according to the expression that $\delta\vec{I} \leftarrow F \cdot (\vec{I_{r1}} - \vec{I_{r2}})$, where the mutation factor F scales the effect of the pairs of chosen individuals on the calculation of the mutation value. Then the mutant individual $\vec{I_m}$ is produced via modifying each gene of $\vec{I_{r3}}$ with the $\delta\vec{I}$, which is formalized as $\vec{I_m} \leftarrow \vec{I_{r3}} + \delta\vec{I}$. DE exploits binary crossover operation to obtain the trial individual and so that keeps the diversity of population. The trial individual vector $\vec{I_{tr}}$ is generated via crossing $\vec{I_t}$ and $\vec{I_m}$ with the binary crossover scheme as the expression that $\vec{I_{tr}} \leftarrow binCrossover(\vec{I_t}, \vec{I_m}, P)$. The crossover probability, $P \in [0, 1]$, controls the effect of parents on the generation of offsprings. The process of DE algorithm is ended at comparing $\vec{I_t}$ against the new individual $\vec{I_{tr}}$ in terms of fitness and determining whether replace it with the $\vec{I_{tr}}$ accordingly. The better individual will be saved in the position of original $\vec{I_t}$ which is described as,

$$\tilde{\mathbf{I}}_\mathbf{t} = \begin{cases} \vec{I_{tr}} & if \quad f(\vec{I_{tr}}) \leq f(\vec{I_t}) \\ \vec{I_t} & otherwise \end{cases} \tag{1}$$

where f($\vec{I}$) is the objective function of vector $\vec{I}$ to be minimized. For each individual, the above process is repeated parallelly with the max iteration (i.e., generations) of G during evolution. Finally, the individual $\vec{I^*}$ with the best fitness is returned as the optimized result of the DE algorithm.

In this work, Pearson correlation coefficient [33] is taken as the fitness of each individual to evaluate the quality of each individual. This correlation, ρ_{xy}, is calculated as follows:

$$\rho_{xy} = \frac{Cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} = \frac{E(xy) - E(x)E(y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{2}$$

where the numerator is covariance of variable x and variable y, $E(x)$ refers to the expectation of variable x. The denominator is the product of the standard deviations of variable x and variable y.

The correlation is used to compare computational results of various similarity methods with the human judgments for word pairs. It is a floating point value between -1 (extreme negative correlation) and +1 (extreme positive correlation) which indicates the degree of linear dependence between the computational methods and human opinion. The nearer the value of correlation is to any of the extreme values (-1 or +1), the stronger is the correlation between the variables and the higher is the performance of the method. If the Pearson correlation of a method gets near to 0, it indicates the method results in poor performance. In terms of Pearson correlation, we compare the performance of our combination strategy and other methods for semantic similarity measurement. Besides, the parameters of DE algorithm consisting of NP, F, P and G need to be fixed as constants. In the following Section 4, we give the concrete values conducted in our experiments.

4 Experiments and results

In this section we demonstrate the experiments which conduct the combination of various vector similarity metrics on different benchmarks and discuss the results. In order to measure semantic similarity between concepts in continuous feature vector space, we learn continuous distributed concept vectors by training neural network model.

4.1 Methodology

We use the tool word2vec[1] to implement CBOW neural network model since its effectiveness and simplicity. We formalize a refined vocabulary as V. For a word w in V, the CBOW model averages the set of its context c_t ={w_{t-k}, ..., w_{t-1}, w_{t+1}, ..., w_{t+k}} which consists of k words to the left and right at projection layer. The training objective of CBOW is to maximize the log probability of the target word w, formally,

$$Obj = \frac{1}{T}\sum_{t=1}^{T} \sum_{(-k \leq j \leq k, j \neq 0)} logp(w_t | w_{t+j}) \tag{3}$$

where w_t is a given target word, w_{t+j} is the surrounding words in context, and k is the context window size. The inner summation spans from -k to +k to compute the log probability of correctly predicting the central word w_t given all the context words w_{t+j}. The conditional probability $p(w_t|w_{t+j})$ is defined in the following softmax function:

$$p(w_t | w_{t+j}) = \frac{exp(vec'(w_t)^\top vec(w_{t+j})}{\sum_{w=1}^{V} exp(vec'(w)^\top vec(w_{t+j}))} \tag{4}$$

where $vec(w)$ and $vec(w`)$ refer to the input vector and output vector of word w.

Three unlabeled corpora are fed as input of the CBOW model, including Wikipedia[2] (3,483,254 word types and 10^9 tokens), BNC[3] (346,592 word types, 10^7 tokens) and Brown Corpus[4] (14,783 types, 10^5 tokens). Once the input corpora are available, pre-processing of corpus is conducted firstly, including data cleaning, tokenization, abbreviation removal, stop-word removal, etc. Named entities and

[1] http://word2vec.googlecode.com/svn/trunk/

[2] http://dumps.wikimedia.org/enwiki/20140903/

[3] http://www.ota.ox.ac.uk/desc/2554

[4] http://nltk.googlecode.com/svn/trunk/nltk_data/index.xml/

special terms that contain uppercase letters are taken as abbreviations and removed from the corpus since they may significantly impact the training precision. In most studies on NLP, stop words are considered useful for handling syntax information, such as progressive relationship and transition relation. However, we consider that this work mainly focuses on the expression ability of word vectors, whereas stop words which occur frequently disturb the sense-group of sentences due to they have little real meaning. Therefore, the stop words are removed to avoid over-training and make the remaining lexical meaning clearly represented. Therefore, we get a vocabulary of over 0.8 billion tokens after processing the raw corpora in advance.

Based on the generated continuous vectors, different similarity results between concepts are computed by various vector metrics. These results are input into the DE algorithm to obtain a optimized value. Table 3 summarizes the configuration settings of the DE algorithm in this work, which provides more competitive results based on the $rand/1/bin$ strategy than other variants of DE algorithm[5].

Table 3. Optimal parameters.

Parameter	Value
Population size, NP	10*N
Mutation factor, F	0.5
Crossover probability, P	0.1
Max generations, G	1000
Max, Min	+10, -10

4.2 Benchmark datasets

7 benchmarks are conducted in our experiments for results verify, including WS-353, WS-sim, WS-rel, RG-65, MC-30, YP-130 and MTurk-287. These datasets are widely used in word similarity studies to compare the semantic similarity methods with human judgements. The **WS-353** dataset [14] contains of 353 word pairs of English words with similarity rating by humans. The degree of similarity of each pair is assessed on a scale of 0-10 by 13-16 human subjects, where the mean is used as the final score. WS-353 was further divided into two subsets [1] that similar pairs (**WS-sim**) and related pairs (**WS-rel**) in terms of the degree of similarity between word pairs. The **RG-65** [32] contains 65 pairs of words assessed on a 0-4 scale by 51 human subjects. The **MC-30** dataset [25], 30 word pairs from RG-65, are reassessed by 38 subjects and a small portion of WS-353. Although these datasets contain overlapping word pairs, their similarity scores are different since they are given by different human judges in the diverse experiments. In addition, the WS-353 contains the words within various part-of-speeches whereas others merely contain nouns. We also evaluate our model on the **Mturk-287** benchmark [31] which consists of 287 word pairs evaluated by 10 subjects on a scale of 1 to 5 for each and crowdsourced from Amazon Mechanical Turk. To specifically emphasize the effect on verb, the **YP-130** dataset [39] that contains 130 verb pairs was created and judged by human as well.

[5]http://www1.icsi.berkeley.edu/ storn/code.html

4.3 Result discussion

We conduct three kinds of experiments to evaluate the proposed approach described in Section 3. Firstly, we compare our DE-based approach with two different sets of similarity metric (vector-based metrics and WordNet-based metrics) on the RG-65 benchmark dataset. Next, we implement our approach on multiple benchmark datasets. Finally, we investigate the parameters of CBOW model which include dimension and window size to demonstrate the robustness of our approach and the effect of these parameters on the similarity measurement of concepts.

4.3.1 Experiments with different metrics on RG dataset

Our approach is compared against two sets of metrics on RG dataset. Firstly, we evaluate various similarity metrics based on the continuous vectors extracted from corpus. Table 4 presents the Pearson correlation between the computational results and human ratings on RG dataset, where the top lists the performance of individual similarity metrics and the bottom shows the result of our DE-based approach. The experimental results demonstrate that our approach improves the accuracy of existing corpus-based vector similarity metrics and achieves a result of 0.894 with the dimension of 500 and window size of 7. While the result of cosine metric which is considered as most effective in most of previous literatures achieves 0.805.

Table 4. Pearson correlation between computational vector metrics and human ratings on RG dataset.

Similarity method	Correlation
Chebyshev	0.660
Tanimoto	0.785
Manhattan	0.788
Euclidean	0.794
Correlation	0.805
Cosine	0.805
Ours (6 metrics)	**0.894**

In order to take full advantage of the semantic information from both WordNet and corpus, we further integrate two additional gloss-based methods into the DE strategy. As mentioned in Section 1, WordNet-based similarity methods contain four categories that path-based, IC-based, feature-based and hybrid methods. In this experiment, we focus on the feature-based methods where the feature properties of WordNet are used to construct concept vectors. Therefore, beside the vector metrics presented in Table 4, our approach combines extended gloss overlap [6] and cosine similarity of gloss vector [28]. For comparison, we choose some hybrid methods which tend to be superior to other WordNet-based methods since they adequately employ various semantic information from WordNet.

Table 5. Pearson correlation between WordNet-based similarity methods and human ratings on RG dataset.

Similarity method	Correlation
Extended gloss overlap[6]	0.350
Gloss vector[28]	0.797
Liu [21]	0.810
Pirro[30]	0.872
Gao[16]	0.885
Ours (8 metrics)	**0.903**

Table 5 indicates that our DE-based combination better aligns with human judgement in contrast with the individual feature-based methods and hybrid methods in the studies related to WordNet. The results also show that continuous vectors learned from corpus seem to supply more precise semantic than the gloss vector extracted from WordNet. Moreover, although having relatively high performance as well as our approach, the hybrid method proposed by Gao [16] requires parameters to be settled.

4.3.2 Experiments with different datasets

Table 6 summarizes the results of state-of-the-art similarity methods on 7 benchmarks, such as WS-353, YP-130, etc. While outperforming our approach on the WS-353, WS-sim and WS-rel dataset, the approach of Yih [35] needs more heterogeneous semantic sources (web search, Wikipedia, Bloomsbury and WordNet) to turn out averaged cosine similarity score. Based on both web corpus and WordNet, Agirre et al. [1] conduct a supervised combination of several similarity methods, which obtains a higher result than ours on the RG-65 dataset. However, their approach has to train a SVM to turn parameters and needs a mass of training data. Unlike some approaches [31] that perform well on some datasets but poorly on others, our approach is more robust since it holds high performance on the additional MTurk-287 dataset and YP-130 dataset. In order to further evaluate the quality of the continuous real-value vectors learned via neural network training, we perform our DE-based approach across different parameter settings.

4.3.3 Experiments with different parameters of CBOW model

In our study, the quality of concept vector depends on the hyper-parameters of CBOW model. To further indicate the robustness of our approach, we estimate the window size of training and the dimensionality size of vector. The window size is set to 3 up to 9. The dimensionality which reveals the feature granularity of vectors ranges from 100 to 900 with a step length of 100. According to the results on RG

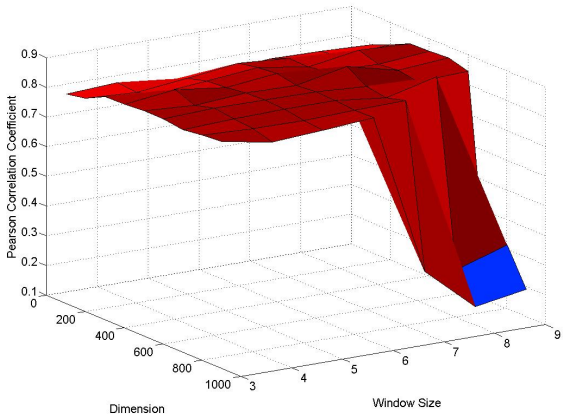

Figure 3. The performances of our method under different settings of dimensionality and window.

dataset shown in Fig. 3, our approach keeps steady across different dimensionality and window sizes, which implies the continuous vector representations used in our approach remain stable expression of semantic features. However, the curved surface suffers a drastic decline near the point with dimensionality 900 and window 9 due to the overfitting resulted by excessive training.

5 Conclusions

This work proposes a differential evolutionary based approach to measure the semantic similarity in a continuous vector space. The differential evolutionary algorithm is used to leverage the results derived from different vector-based similarity metrics and find a optimal combination strategy of the metrics. The continuous vectors which reveal

Table 6. The performance of state-of-the-art methods on multiple datasets.

Similarity method	RG-65	MC-30	WS-353	WS-sim	WS-rel	MTurk-287	YP-130
Yih [35]	0.89	0.89	**0.81**	**0.87**	**0.77**	0.68	NA*
Radinsky [31]	NA	NA	0.80	NA	NA	0.63	NA
Agirre [1]	**0.96**	0.92	0.78	0.83	0.72	NA	NA
Ours (8 metrics)	0.90	**0.93**	0.76	0.83	0.70	**0.71**	**0.75**

* N/A means empty value.

latent semantic features of words are explored to improve the vector similarity computation. The experiment results demonstrate our combined approach outperforms other similarity methods on multiple benchmark datasets and has the robustness under different training parameters. In future works, we will present an WordNet-constrained neural network model to further improve the quality the distributed vectors and the accuracy of the semantic similarity measurement between concepts.

Acknowledgment

This work is supported in part by National Natural Science Foundation of China (No.61272353, 61370128, and 61428201), Program for New Century Excellent Talents in University (NCET-13-0659), Beijing Higher Education Young Elite Teacher Project (YETP0583).

References

[1] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of the conference of The North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pages 19–27, June 2009.

[2] A. O. Alves, A. Ferrugento, M. Lourenço, and F. Rodrigues. Asap: Automatic semantic alignment for phrases. In *the 8th International Workshop on Semantic Evaluation (SemEval)*, pages 104–108, Dublin, Ireland, August 2014.

[3] B. V. Babu and S. A. Munawar. Differential evolution strategies for optimal design of shell-and-tube heat exchangers. *Chemical Engineering Science*, 62:3720–3739, 2007.

[4] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 2200–2204. European Language Resources Association, May 2010.

[5] C. Banea, D. Chen, R. Mihalcea, C. Cardie, and J. Wiebe. Simcompass: Using deep learning word embeddings to assess cross-level similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval)*, pages 560–565, August 2014.

[6] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *International Joint Conference on Artificial Intelligence*, volume 3, pages 805–810, August 2003.

[7] D. Bär, C. Biemann, I. Gurevych, and T. Zesch. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440. Association for Computational Linguistics, June 2012.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, January 2003.

[9] J. M. Chaves-González and J. MartíNez-Gil. Evolutionary algorithm based on different semantic similarity functions for synonym recognition in the biomedical domain. *Knowledge-Based Systems*, 37:62–69, January 2013.

[10] Y. Chen, B. Perozzi, R. Al-Rfou, and S. Skiena. The expressive power of word embeddings. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, June 2013.

[11] S. Das and P. N. Suganthan. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation*, 15(1):4–31, 2011.

[12] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, September 1990.

[13] M. Faruqui and C. Dyer. Community evaluation and exchange of word vectors at wordvectors.org. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, June 2014.

[14] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131, January 2002.

[15] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3:1289–1305, March 2003.

[16] J.-B. Gao, B.-W. Zhang, and X.-H. Chen. A wordnet-based semantic similarity measurement combining edge-counting and information content theory. *Engineering Applications of Artificial Intelligence*, 39:80–88, 2015.

[17] F. Hill, D. Kiela, and A. Korhonen. Concreteness and corpora: A theoretical and practical analysis. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 75–83. Association for Computational Linguistics, August 2013.

[18] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 873–882, Jeju Island, South Korea, July 2012.

[19] D. Kiela and S. Clark. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) at EACL*, pages 21–30. Association for Computational Linguistics, April 2014.

[20] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, pages 296–304, Madison, Wisconsin, USA, July 1998.

[21] H. Z. Liu, H. Bao, and D. Xu. Concept vector for semantic similarity and relatedness based on wordnet structure. *The Journal of Systems and Software*, 85(2):370–381, August 2012.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *the International Conference on Learning Representations (ICLR) Workshop*, Scottsdale, Arizona, USA, May 2013.

[23] T. Mikolov, W. tau Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *the conference of North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pages 746–751, Atlanta, GA, USA, June 2013.

[24] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, November 1995.

[25] G. A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

[26] E. nn Mezura-Montes, J. Velázquez-Reyes, and C. A. C. Coello. A comparative study of differential evolution variants for global optimization. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 485–492, New York, NY, USA, 2006. ACM.

[27] A. V. Paliwal, B. Shafiq, J. Vaidya, H. Xiong, and N. Adam. Semantics-based automated service discovery. *IEEE Transactions on Services Computing*, 5(2):260–275, May 2012.

[28] S. Patwardhan and T. Pedersen. Using wordnet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL Workshop on Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8. Citeseer, March 2006.

[29] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, October 2014.

[30] G. Pirró. A semantic similarity metric combining features and intrinsic information content. *Data & Knowledge Engineering*, 68(11):1289–1308, 2009.

[31] K. Radinsky, E. Agichtein, E. Gabrilovich, and S. Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on world wide web*, pages 337–346. ACM, March 2011.

[32] H. Rubenstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October 1965.

[33] J. S. Simonoff. *Smoothing methods in statistics*. Springer, 1996.

[34] R. Storn and K. Price. Differential evolution- a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[35] W. tau Yih and V. Qazvinian. Measuring word relatedness using heterogeneous vector space models. In *the conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies*, pages 616–620. Association for Computational Linguistics, June 2012.

[36] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, July 2010.

[37] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1):141–188, February 2010.

[38] E. M. Voorhees. Query expansion using lexical-semantic relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference*, pages 61–69. Springer, January 1994.

[39] D. Yang and D. M. Powers. Verb similarity on the taxonomy of wordnet. In *Proceedings of the 3rd International WordNet Conference (GWC)*, pages 121–128, January 2006.