A Synonym Extraction Method Based on Intimacy*

1st Ru Wang, 2nd Wei Pan , 3rd JingHui Ma , 4th HangXing Wang
Beijing Key Laboratory of Electronic System Reliability and Prognostics
Capital Normal University
Beijing, China
wangru158105@163.com

***Abstract*—This paper proposes a new concept—intimacy for synonyms extraction.Firstly, we find the first n similar word labeled A1...An of the word A, and then, find the first n similar word groups of A1,A2...An respectively, and calculate the intimacy based on whether or not A appears in these word groups and the position A appears to obtain the final intimacy score. Finally, sort the final intimacy score and obtain the first n synonyms of A. In this paper, we test on Wikipedia dataset, and compare with word similarity computing tasks. Experiments show that the synonyms found by this method are higher than other methods both in subjective and objective evaluation.**

***Index Terms*—Word Similarity, Synonym Extraction, Intimacy.**

I. Introduction

With the rapid development of the Internet, the information shows exponential growth in network media, which means how to make full use of it is particularly significant in the fields of Natural Language Processing(NLP), data mining and information retrieval. Synonym extraction, as a basic job, is important to many natural language processing applications such as question answering[2,3], web searching[4] and so on. In traditional NLP systems, every word is an index in the vocabulary, and there is no correlation between words[5].At present there is no rigorous definition of synonyms. Bengio et al[6] put forward "word embedding" for the first time in 2003. Since the words can be vectorized so they can be quantitatively measured, then the similarity between words can be compared directly with a certain kind of method, for example, the cosine distance[7]. After the word embedding was proposed, a number of word embedding systems are emerged such as: Latent Dirichlet Allocation(LDA)[8,9],word2vec[10,11], and GloVe [15]. Among them, LDA is a generative statistical model, it uses *multinomial distribution* to explain the relationship between documents, topics, or words, and at the same time it needs to be combined with Latent Semantic Allocation (LSA) to get word embedding. Word2vec[10,11], based on the neural networks (skip-gram, Continuous Bag-of-Words), obtains word embedding by learning and training corpus. But sometimes, the similarities between a word and its antonyms are also very close, for example, "good" and "bad", the cosine similarity is 0.727(When the two words are completely similar, the cosine similarity is 1), higher than its synonym "excellent":0.597, and this may lead to inconsistent results. GloVe can get the linear substructures of the word vector space by constructing the co-occurrence matrix. However, with the expansion of the datasets, for example, the Wikipedia corpus is about 13G, the training process will become more difficult and spend more time. In order to calculate the similarity between two words and obtain synonyms,Dmitry et al[17,18] proposed a graph-based approach which can induce synsets using synonymy dictionaries and word embedding, as same as web search algorithm[13]. However, the method depends on the structures of the input synonymy graphs, the sparse or dense input of the synonymous graph directly affects the result of the calculation. Nazar et al[21] presented an approach to compute the similarity of English word pairs but not individual word. In this work they used relational features between words, however, they didnt use attributional similarity. Andrew and Alexander[13] thought that the notion of synset(a set of synonyms) owes its occurrence to the WordNet[14], where different relations (synonymy, homonymy) are indicated between synsets but not between individual words[20]. However, with the increase of new words or the emergence of new meanings of words, the results are difficult to control. In conclusion, if there is a method that can analyze the relationship between synonyms more appropriately and quantitatively, then the task of synonym extraction will be easier. This paper proposes a new concept of intimacy to measure the similarity between words,called the IBW(Intimacy between words). It is based on people's perception of intimacy. In general, if both pay attention to each other, their intimacy will be higher than if only one of them pays attention and the other does not. Suppose there is a seed word A, firstly, we find the first n similar words labeled A1...An of the word A, and then,find the first n similar word groups of A1,A2...An respectively, and calculate the intimacy based on whether or not A appears in these word groups and the position A appears, these will be used to calculate the final intimacy score. Finally, sort all these words and determine the new n words as the synonyms of A. The experiments prove that the similarity of the words found in this method is more realistic than other methods.

II. Related Work

As one of the most famous word embedding systems, the word2vec provides the state-of-the-art results on linguistic

This work was supported by the National Key R & D Plan (2017YFC0806700),National Natural Science Foundation of China (61876111,61702348,61772351), and Capacity Building for Sci–Tech Innovation – Fundamental Scientific Research Funds (025185305000).

tasks, it is released by Google in 2013. Using the skip-gram model[7] and Continuous Bag-of-Words(CBOW) model[11] to train and obtain the word embedding. Although both of them contain three layers: input layer, hidden layer and output layer, their input and output are different. For example, suppose there is a sentence consists five words s(t-2), s(t-1), s(t),s(t+1),s(t+2), the CBOW predicts the current word s(t) according to the surrounding context words s(t-2), s(t-1), s(t+1),s(t+2), while the skip-gram predicts the surrounding context words s(t-2), s(t-1), s(t+1),s(t+2) according to s(t)[12,16]. The word2vec uses cosine distance to compare the similarity between words, but sometimes the results are not satisfied, for example, by using the word2vec we can get 16 words similar to "bird" in Table I.

TABLE I
TOP 16 SIMILAR WORDS OR SYNONYMS FOR BIRD.

Birds	0.733
Parrot	0.644
Mynah	0.617
Turtle	0.601
Owl	0.592
Feather	0.589
Lorikeets	0.564
Taxidermied	0.558
Magpie	0.580
Ptitsy	0.542
Chirping	0.552
Parakeets	0.549
Foodle	0.542
Gooney	0.534
Dodo	0.518
Fowl	0.546

It can be found that "turtle" and "feather" is not belong to birds, but their rankings are very high, even exceed the real birds: "owl". Several methods are used to solve this problem[11,16,17,20,21], but the results are not satisfied. In this paper, we propose a new method named intimacy to solve this problem.

III. THE CONCEPT OF INTIMACY

In this paper, the "intimacy" means: the similarity between individuals, which is used to describe the selection criteria for candidate words. This is not only the similarity between two words, but also between multiple words, for example, in Table II, we can get the similar words of "owl", "turtle" and "feather" by using the word2vec :

It can be found that both "turtle" and "owl" appear in the similar words of "bird", and their rankings are also high, however, "bird" in the similarity ranking of "owl" clearly exceeds the position of "bird" in "turtle". Obviously, "bird" and "owl" are more similar than "bird" and "turtle". Similarly, "feather" appears in the similarity ranking of "bird", but "bird" doesn't appear in the similarity ranking of "feather", therefore, "bird" and "owl" are more similar than "bird" and "feather". This is the theoretical basis of the "intimacy" proposed in this paper, it comes from the intimacy between people.The intimacy of the words Di and Dj is defined as:

TABLE II
THE SIMILARITIES IN DIFFERENT WORDS.

owl	**turtle**	**feather**
Squirrel 0.591	Tortoise 0.704	Feathered 0.625
Bird 0.592	Crab 0.679	Pennaceous 0.620
Alligator 0.525	Snake 0.636	Ruffling 0.596
Glimfeather 0.519	Crocodile 0.587	Ruffle 0.551
Baboon 0.541	Elephant 0.573	Tassle 0.574
Monkey 0.548	Lizard 0.576	Ruffled 0.631
Cat 0.522	Monkey 0.535	Forelock 0.528
Rabbit 0.542	Triggerfish 0.537	Frilled 0.558
Tortoise 0.516	Gourami 0.585	Headpiece 0.551
Parrot 0.589	Parrot 0.564	Plumed 0.595
Elephant 0.501	Alligator 0.661	Vaned 0.554
Lizard 0.497	Shark 0.559	Frill 0.537
Turtle 0.498	Eel 0.540	Kerchief 0.530
Ostrich 0.517	Squirrel 0.548	Matted 0.529
Spectacled 0.555	Toad 0.563	Topknot 0.529
	Ostrich 0.530	Frilly 0.544
	Bird 0.601	Whiskers 0.546

$$S(Di,Dj) = Sim(Di,Dj) + Sim(Dj,Di) + \sum_{m=1}^{n} Sim(Di,Djm) + \sum_{m=1}^{n} Sim(Dj,Dim) \quad (1)$$

S is the intimacy fraction, D is the word set similar to "bird", n is the word groups of Dj and Di, n is an adjustable parameter, We find that when n is 117, it works best. The position function is also called the sigmoid function,the reason we chose it is: 1. The sigmoid function converges within (0,1) . 2. The closer to the target word, the higher the fraction. It is defined as :

$$Sim(Di,Dj) = \frac{1}{1+e^{-x}} \quad (2)$$

Where x is associated with the position.Table III shows the results after calculating the intimacy of bird. We list the top 5 intimacy words after calculation(the IBW). It shows that the top 5 intimacy words are all belong to "bird".

TABLE III
THE INTIMACY USED IN THE IBW.

Seed words	**bird**
the IBW	Birds
	Jubjub
	Owl
	Mynah
	Ostrich

IV. METHODOLOGY

We obey the following experimental steps:

- Obtain the first n similar words("Birds","Mynah",..."Tortoise") of the "bird".
- Find the first n similar word groups of the words in step1 respectively .
- Calculate the fraction of intimacy according to the position "bird" appears in the words of step 1 and word groups of step 2.

- Sum and resort the fraction of each word above according to intimacy to get the final result.

In addition to above, during the preliminary processing of wordset, we also use NLPIR participle system system to remove words from different parts of speech.

V. Experiments

A. Training Models

In this paper we used Wikipedia corpus to train word2vec and Glove,as a comparison, during training word2vec,we set the dimension is 200, the window is 5 and min_count is 10.

B. Analysis And Comparison Of Experimental Results

To evaluate the validity of our model, we compared our results with the word2vec, the WordNet and Glove. We selected about 103 most common seed words to test, and asked 11 persons to give their marks to the words, if they thought the word is exactly similar to seed word, the score is 10, if they thought they have no similarity, the score is 0. Due to the limitation of the length of this article, Table IV and Table V list the top 7 similar words of each method. If the number of synonyms are less than 7,then all are listed. The last two lines show the average mark and standard deviation. Table VI shows the average of 103 common seed words average mark and standard deviation.

TABLE IV
SYNONYMS OF "BIRD".

Bird				
	IBW	word2vec(wiki)	WordNet	Glove(wiki)
	Birds	Birds	Bird	Birds
	Jubjub	Parrot	Fowl	melaselvanur
	Owl	Mynah		kersees
	Mynah	Turtle		hohow
	Ostrich	Owl		attiveri
	Goose	Feather		gerhl
	Dodo	Lorikeets		mangalavanam
ave	7.64	7.01	8.22	4.04
std	1.97	2.06	1.73	4.13

TABLE V
SYNONYMS OF "BEIJING"

Beijing				
	IBW	word2vec(wiki)	WordNet	Glove(wiki)
	Shanghai	Shanghai	Beijing	Sineirina
	Guangzhou	Taibei		Shanghai
	Shenzhen	Shenzhen		China
	Taipei	Seoul		Zaoxing
	Nanjing	China		Guangzhou
	Tianjin	Changchun		Taolunhui
	Wuhan	Baxy		seoul
ave	7.11	6.83	10	5.13
std	2.03	2.47	0	4.51

Table IV, Table V and Table VI shows that the IBW is more close to human estimates than other methods, we can get the same effect and even better than those methods: For "bird", the word2vec expands "turtle" , it is obviously not belong to birds.

TABLE VI
AVERAGE OF 103 COMMON SEED WORDS

method	**IBW**	**word2vec(wiki)**	**WordNet**	**Glove(wiki)**
Average of ave	7.03	6.77	7.71	4.72
Average of std	2.04	2.52	1.81	4.01

For "Beijing", our method expands "Shanghai, Guangzhou", however, the WordNet only find "Beijing", which is less rich due to the fact that it focus on a particular aspect[18].Glove only focus on cooccurence, it contains limited semantic information for the word embeddings,resulting in a lower score.

C. Evaluation Using F score

We calculated the F score of top 10 and top 20 of rank lists in our results. The F score is used in the field of information retrieval and statistical classification to evaluate the quality of the results. In this paper it is known as F[25,26]:

$$F = 2 \bullet \frac{precision \bullet recall}{precision + recall} \tag{3}$$

Precision is used to calculate the degree of the evaluation words(words in the WordNet synonym dictionary) in the top 10 of expanded word list, recall is used to calculate the degree of the evaluation words expanded by models in the top 10 of word list. Higher values indicate better results. We selected about 20K most common seed words and compared our results with one well-known embedding, Glove[15], and we validated our model(the IBW) on the Wikipedia data(wiki) the results are given in Table VII.

TABLE VII
EVALUATION OF OUR METHOD AND GLOVE.

models	F(10)	F(20)
IBW(wiki)	**0.339**	**0.370**
word2vec(wiki)	0.338	0.363
Glove(wiki)	0.234	0.245

In Table VII, the F score of ours is the highest compare with the word2vec(wiki) and Glove(wiki),which proves the effectiveness of our method.

D. Evaluation Using ρ

Above all, in order to prove that the algorithm is closer to human cognition, we also reported the spearmans rank correlation coefficient. It assesses the relationship between two variables, and it can be described by using a monotonic function. A perfect spearman correlation of +1 or -1 occurs when each of the variables is a perfect monotone function of the other. It is defined as:

$$\rho_{r_x, r_y} = \frac{cov(r_x, r_y)}{\sigma_{r_x} \sigma_{r_y}} \tag{4}$$

Where

ρ denotes the usual Pearson correlation coefficient, but applied to the rank variables.

$cov(r_x, r_y)$ is the covariance of the rank variables.

σ_{r_x} and σ_{r_y} are the standard deviations of the rank variables.

r_x and r_y are the ranks of x,y.

We used several different datasets: WordSim353[27], SCWS[28], Rare Words[24], MEN[29] and SimLex999[30], which provide words and their similarity judgments by humans. We selected about 20k most common used seed words and used the word2vec and the Glove as comparison. Table VIII presents the (average) Spearman rank correlation for various datasets.

TABLE VIII
TABLE TYPE STYLES

Datasets	**WordSim353**	**SCWS**	**Rare Words**	**MEN**	**SimLex999**
word2vec(wiki)	0.493	0.532	**0.323**	0.531	0.215
IBW	**0.514**	**0.537**	0.317	**0.540**	**0.224**
Glove(wiki)	0.304	0.206	0.148	0.224	0.054

It can be found that our model outperforms the word2vec(wiki) and Glove(wiki). The words in Rare Words are rare and complex, causing interference in the training of the words and resulting in a lower score.Therefore, it is necessary to pay attention to the training of complex and rare words.

VI. CONCLUSION

This paper proposes a new method of synonym extraction, it is based on people's perception of intimacy, and it is different from traditional similarity-based methods. Firstly, we find the first n similar words labeled A1...An of the word A, and then,find the first n similar word groups of A1,A2...An respectively, and calculate the intimacy based on whether or not A appears in these word groups and the position A appears, these will be used to calculate the final intimacy score. Finally, sort all these words and determine the new n words as the synonyms of A. Experiments show that our method outperforms the other methods both in subjective and objective evaluation.

REFERENCES

[1] Zhiguo Gong, Chan Wa Cheang, and U. Leong Hou. Web Query Expansion by WordNet[C].In Proceedings of the 16th International Conference on Database and Expert Systems Applications - DEXA05, Springer Berlin Heidelberg, Copenhagen, Denmark, pages 166–175.2005.https://doi.org/10.1007/11546924 17.

[2] Kwok C , Etzioni O , Weld D S . Scaling question answering to the web[J]. ACM Transactions on Information Systems, 19(3):242–262,2001.

[3] Zhou G , Liu Y , Liu F , et al. Improving Question Retrieval in Community Question Answering Using World Knowledge[J]. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. AAAI Press, Beijing, China, IJCAI'13, pages 2239 – 2245.2013.

[4] VD Blondel , A Gajardo, M Heymans, P Senellart, PV Dooren. A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching. Siam Review, 46 (4) :647–666. 2004 .

[5] Keet Sugathadasa, Buddhi Ayesha. Synergistic Union of the word2vec and Lexicon for Domain Specific Semantic Similarity.arXiv print arXiv:1706.01967v2. 2017.

[6] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. Journal of Machine Learning Research 3:1137 –1155 . 2003

[7] Y. Goldberg and O. Levy, the word2vec explained: Deriving mikolov et al.s negative-sampling word-embedding method, arXiv preprint arXiv:1402.3722, 2014.

[8] R. Das, M. Zaheer, and C. Dyer, " Gaussian lda for topic models withword embedding." in ACL (1), pp. 795-804.2015.

[9] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022. 2003.

[10] Tomas, Kai Chen, Greg Corr, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 . 2013.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality[J]. In Advances in neural information processing systems. 26:3111–3119. 2013.

[12] Lian Z . Exploration of the Working Principle and Application of Word2vec[J]. Sci-Tech Information Development & Economy, 2015.

[13] Hugo Caselles-Dupr, Florian Lesaint, Jimena Royo-Letelier, the word2vec applied to Recommendation: Hyperparameters Matter.arXiv print arXiv:1804.04212v2. 2018.

[14] Miller G A , Beckwith R , Fellbaum C , et al. Introduction to the WordNet: An On-line Lexical Database[J]. International Journal of Lexicography, 3(4):235–244.1990.

[15] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),pages 1532–1543. 2014.

[16] Mikolov T , Chen K , Corrado G , et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[17] D Ustalov, A Panchenko, C Biemann. Automatic Induction of Synsets from a Graph of Synonyms. Meeting of the Association for Computational Linguistics , 2017 :1579–1590,2017.

[18] D Ustalov , M Chernoskutov, C Biemann , A Panchenko. Fighting with the Sparsity of Synonymy Dictionaries for Automatic Synset Induction. International Conference on Analysis of Images , 2017 :94–105,2017.

[19] O. Melamud, O. Levy, I. Dagan, and I. Ramat-Gan, "A simple word embedding model for lexical substitution," in Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 1–7. 2015.

[20] Andrew Krizhanovsky, Alexander Kirillov, 2018. Calculated Attributes of Synonym Sets. arXiv print arXiv:1803.01580v1. 2018

[21] Khan N , Shaukat A . New Word Pair Level Embeddings to Improve Word Pair Similarity[C]// Iapr International Conference on Document Analysis & Recognition. IEEE, 2018.

[22] Ustalov D , Panchenko A , Biemann C . Watset: Automatic Induction of Synsets from a Graph of Synonyms[J]. 2017.

[23] Zhang L , Li J , Wang C . Automatic synonym extraction using Word2Vec and spectral clustering[C]// 2017 36th Chinese Control Conference (CCC). IEEE, 2017.

[24] Minh-Thang Luong, Richard Socher, and Christopher D Manning. Better word representations with recursive neural networks for morphology. CoNLL-2013, 104. 2013.

[25] Maria Pelevina, Nikolay Arefiev, Chris Biemann, and Alexander Panchenko. Making Sense ofWord embedding. In Proceedings of the 1st Workshop on Representation Learning for NLP. Association for Computational Linguistics, Berlin, Germany, pages 174–183. 2016.

[26] Hope D, Keller B. MaxMax: A Graph-Based Soft Clustering Algorithm Applied to Word Sense Induction[J]. 2013.

[27] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin, "Placing Search in Context: The Concept Revisited", ACM Transactions on Information Systems, 20(1):116–131. 2002.

[28] Huang, Eric H, Richard Socher, Christopher D Manning, and Andrew Ng. Improving Word Representations via Global Context and Multiple Word Prototypes. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, 873–82. 2012.

[29] Bruni, Elia, Nam Khanh Tran, and Marco Baroni. Multimodal Distributional Semantics. Journal of Artificial Intelligence Research 49: 1–47. 2014

[30] Hill, Felix, Roi Reichart, and Anna Korhonen. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. Computational Linguistics 41 (4): 665–95.2015.