Semantic Web 0 (0) 1
IOS Press

Wan2Vec: Embeddings Learned on Word Association Norms

Gemma Bel-Enguix [a], Helena Gómez-Adorno [a,*], Jorge Reyes-Magaña [a,b] and Gerardo Sierra [a]
[a] *Instituto de Ingeniería (II), Universidad Nacional Autónoma de México (UNAM), Mexico City, Mexico*
E-mails: hgomeza@iingen.unam.mx, gbele@iingen.unam.mx, jorge.reyes@correo.uady.mx
[b] *Facultad de Matemáticas (FM), Universidad Autónoma de Yucatán (UAdY), Merida-Yucatan, Mexico*
E-mail: jorge.reyes@correo.uady.mx

Abstract. Word embeddings are powerful for many tasks in natural language processing. In this work, we learn word embeddings using weighted graphs from Word Association Norms with the *node2vec* algorithm. The computational resources used by this technique are reasonable and affordable, which allows us to obtain good quality word embeddings even from small corpus. We evaluate our word vectors in two word similarity benchmarks, the *WordSim-353*, *MC30*, *MTurk-287*, *MEN-TR-3k*, *SimLex-999*, *MTurk-771* and *RG-65*, achieving better results than those obtained with *word2vec*, *GloVe*, and *FastText*, trained on huge corpus.

Keywords: Word Association Norms, Word Embeddings, Learning

1. Introduction

Semantic representation of words in a vector space has been an active research field over the past decades. Computational models like singular value decomposition (SVD) and latent semantic analysis (LSA) [1] are able to model continuous representations of words (embeddings) from term-document matrices. Both methods are able to reduce an n-dimensional term-document matrix using only the most important dimensions.

More recently, *M*ikolov et al. [2] introduced *word2vec*, inspired by the distributional hypothesis, which states that words in similar contexts tend to have similar meanings [3]. This technique uses a neural network to learn vector representations of words by predicting other words in the neighborhood. Distributed vector representations obtained with *word2vec* have the capability to preserve linear regularities between words.

Some alternative methods have been designed to improve the performance of *word2vec*. *Glove* [4] aims to be an efficient vector model by training nonzero elements in a word-word co-occurrence matrix. *FastText* [5] is an approach based on the skip-gram model, with the difference that in this method every word is represented as a bag of character n-grams. *Mikolov et al.* [6] claim that the models trained with *FastText* exhibit the best degree of accuracy compared to other systems, becoming the new state-of-the-art in distributed representations of words.

In order to build an adequate and reliable semantic vector space model able to capture semantic similarities and linear regularities of words, huge volumes of text are necessary. Although methods for learning word vectors are fast and efficient to train, and pre-trained embeddings are usually available online, it is still computationally expensive to train huge volumes of data in non-commercial environments, i.e., personal computers. Acquisition of semantic representations require large handcrafted knowledge bases or huge volumes of text to be exploited. Some *word2vec* embeddings are trained on billions words, which implies a computational cost that is not very realistic from the cognitive point of view either. In addition to this, only really big sources like Wikipedia, an encyclopedia, can provide the amount of lexical items necessary to perform such processes. Encyclopedias have plenty of terms that are not part of common language (ie., Latin

*Corresponding author. E-mail: hgomeza@iingen.unam.mx.

names for plants and animals, geographical names, etc.), introducing some unneeded noise to the vectors.

Even though millions of different words can be computed for languages like English, this is not the actual size of the vocabulary an English speaker uses in daily life. Psycholinguists do not agree on the number of words that comprise the core of a language. In the thirties of the 20th Century, Ogden [7] suggested that there is a small set of 850 items that constitute what he calls "basic English". Although this theory has been broadly criticized [8], daily life is very limited.

In another vein, methods based on cooccurrences of words, like *word2vec*, provide a good idea on the syntagmatic and paradigmatic relations of words, but fail to show their psychological and associative connections. When envisioning some writing assistants and applications to help with real problems of dysnomia, these latter types of relations cannot be dismissed.

Therefore, it makes sense to look for models that can work with more restricted sets of both, tokens and types. Somehow, these methods can decrease the recall, but the computational complexity and the efficiency will be increased. Moreover, it is very recommended to consider corpora that can provide other information on how the lexicon is structured and accessed, rather than corpora where the relation between words are only based on cooccurrence. Wordnet [9], semantic networks or word association networks are possible sources for these experiments. In this paper, we evaluate the convenience of using word association norms as a basis for learning relations between words.

Word association (WA) tests are an experimental technique for discovering the way that human minds structure knowledge [10]. In free association experiments, a person is asked to respond with the first word that comes to mind in response to a given *stimulus* word. Free WA tests are able to produce rich types of associations that can reflect both semantic and episodic memory contents [11] in the form of general Word Association Norms (WAN).

The goal of this paper is to learn continuous feature representations for nodes (that represent words) in WANs. Our hypothesis is that word embeddings learned from WANs map the organized lexical items arranged in the lexicon to a vector space, thus learning richer representations. *Grover and Leskovec* [12] introduced an algorithm called *node2vec* that is able to learn mappings of nodes to a continuous vector space taking into account the network neighborhoods of nodes. The algorithm performs a biased random walk to explore different neighborhoods in order to capture not only the structural roles of nodes (words) in the network but also the communities they belong to. Our proposal is called *wan2vec*, because vectors are learned from a network built over a WAN, by means of the *node2vec* algorithm.

The rest of the paper is organized as follows. Section 2, discusses some related work. In Section 3, we present a well-known compilation of Word Association Norms, the Edinburgh Association Thesaurus, and we describe the methodological framework for learning word embeddings from the NAP. In Section 4, we present the evaluation of the generated vectors using standard data sets for word similarity in Spanish. Finally, in Section 5 we draw some conclusions and point to possible directions of future work.

2. Related Work

The idea of Word Associations was first proposed by psychologists to tackle illnesses like dementia or aphasia. Within cognitive psychology, *Collins* [13] applied them to simulate memory processes. From psycholinguistics, *Clark* [14] presents free associations as an ability that can reveal some properties of the mechanisms of language.

Linguistics and Psycholinguistics used semantic networks [15], which are graphs relating words [16], not only to study the organization of the vocabulary, but also to approach the structure of knowledge.

Word Association Norms (WAN) are corpora of free association words. One of the first examples is provided by *Kent & Rosanoff* [17], who used this method for comparisons of words, introducing 100 emotionally neutral test words. They conducted the first large scale study with 1000 test subjects, and concluded that there was uniformity in the organization of associations and that people shared stable networks of connections among words [18].

Many languages have corpora of WAN. In the past decades, some other association lists were elaborated with the collaboration of a large number of volunteers. In recent years, the web has become the natural way to get data to build such resources. *Jeux de Mots* provides an example in French[1] [19], whereas the *Small World of Words*[2] deals with nine different languages. The repositories that are collected online have certain limitations, especially the lack of control over who is

[1] http://www.jeuxdemots.org/

[2] https://smallworldofwords.org/.

in fact playing, the linguistic proficiency of the users, and their age, gender or level of studies.

Among the best known resources available on the web for English are the Edinburgh Associative Thesaurus[3] (EAT) [20] and the compilation of *Nelson et al.* [21][4].

Borge-Holthoefer & Arenas [11] introduced a Random Inheritance Model for extracting semantic similarity relations from free association information. The obtained vectors were compared with LSA-based vector representations and the WAS (word association space) model. Their results indicate that RIM can successfully extract word feature vectors from a free association network.

In recent years, *Bel-Enguix et al.* [22] used graph analysis techniques to compute associations from large texts collections. Furthermore, *Garimella et al.* [23] introduced a demographic-aware word association model based on a neural net skip-gram architecture, outperforming generic methods for computing associations that do not take into account writer's demographics.

Sinopalnikova and Smrz [24] presented a methodological framework for building and extending semantic networks with word association thesaurus (WAT), including a comparison of quality and information provided by WAT vs. other language resources. The authors showed that WATs are comparable to balanced text corpus and can replace them, in case of absence of a corpus.

In a recent work by *De Deyne et al.* [25] the authors introduced a spreading activation approach in order to encode a semantic structure from a word association network, specifically part of the *Small World of Words*. The word association-based model was compared with a word embeddings model (*word2vec*) using relatedness and similarity judgments from humans, obtaining an average of 13% of improvement over the *word2vec* model.

In this work, we aim at learning vector representation of words using a resource that compiles word association norms in English [20]. Unlike the work of *De Deyne et al.* [25], we use the complete word association corpus (not only the largest connected component), increasing in this way the coverage of the vocabulary by 5% and maintaining high quality vectors.

[3] http://www.eat.rl.ac.uk/

[4] http://web.usf.edu/FreeAssociation

3. Learning embeddings on the Edinburgh Association Thesaurus WAN

The EAT was mainly collected with undergraduate students from different British universities. Their age ranged from 17 to 22.64% of the participants were males and 36% females. Every informant gave responses for 100 words, and every word was given to 100 participants. The resource was elaborated between 1968 and 1971 and published in 1973. Although it could be said that EAT presents an "old stage" of the language, it is the most complete and balanced WAN elaborated so far.

We used an XML version of the resource[5], prepared by the University of Montreal that consists of 8,211 stimulus words, and 20,445 different words including both, stimuli and responses, being these the nodes of the graph.

The graph representation of the EAT corpus is formally defined by $G = \{V, E, \phi\}$ where:

- $V = \{v_i | i = 1, ..., n\}$ is a finite set of nodes of length n, $V \neq \emptyset$, corresponding to the *stimulus* and its *associates*.
- $E = \{(v_i, v_j) | v_i, v_j \in V, 1 \leqslant i, j \leqslant n\}$, is the set of edges.
- $\phi : E \rightarrow \mathbb{R}$, is a weight function over the edges.

The graph is undirected, so that every stimulus is connected to every associated word without any precedence order.

Two different functions were used to assign weight to the edges: Frequency and Association Strength [26]. Only the former is included in the data provided by the EAT. The latter was calculated from the results of frequency for every word associated to its stimulus.

Frequency (F) Counts the number of occurrences of every *response* associated to its *stimulus* in the whole corpus. In order to use this information, we calculated the inverse frequency (IF), that is defined in the following way: being F the frequency of a given associated word, and ΣF the sum of the frequencies of the words connected to the same *stimulus*, $IF = \Sigma F - F$.

Association Strength (AS) Establishes a relation between the frequency (F) and the number of associations for every stimulus. It is calculated as follows: being F the frequency of a given associated

[5] http://rali.iro.umontreal.ca/rali/?q=en/Textual%20Resources/EAT

word W, and ΣF the sum of the frequencies of the *responses* to the same *stimulus* (the total number of responses), the association strength (AS) of the word W to such *stimulus* is given by the formula:

$$AS_W = \frac{F}{\Sigma F}$$

For our experiments, we used the inverse association strength, IAS in order to prepare the system to work with graph-based algorithms:

$$IAS_W = 1 - \frac{F}{\Sigma F}$$

Note that IF and IAS have been obtained because *node2vec* works giving better values to the shorter paths, or those with a smaller weight, whereas in WAN the higher results in Frequency and Association Strength are denoted with higher scores. In order to have a compatible quantification, the concepts Inverse Frequency and Inverse Association Strength have been introduced.

3.1. Node2vec

The *node2vec* algorithm [12] finds a mapping $f : V \rightarrow \mathbb{R}^d$ transforming the nodes of a graph into vectors of d-dimensions. It defines a network neighborhood $N_s(u) \subset V$ for every node $u \in V$ through a sampling strategy S. The aim of the algorithm is to maximize the probability of observing subsequent nodes in a fixed length random walk.

The sampling strategy designed in *node2vec* enables to explore neighborhoods with flexible biased-random walks. Parameters p and q control the switching between breath-first (BFS) and depth-first (DFS) graph searches. Thus, choosing the right balance allows to preserve both community structure and structural equivalence between nodes in the new vector space.

In this work, we used the implementation available on the website[6] of the *node2vec* project with default values for all parameters. We examined the quality of the vectors with different number of features d and different number of walk lengths l.

[6]http://snap.stanford.edu/node2vec/

4. Word Embeddings Evaluation

There are several evaluation methods for unsupervised embedding techniques [27], categorized as intrinsic and extrinsic. The extrinsic evaluation consists in evaluating the quality of the word vectors in Natural Language Processing (NLP) tasks [28, 29]. The improvement is measured in the performance metric specific to the evaluated task. Intrinsic evaluations test the ability of word vectors to capture syntactic or semantic relationships [30] of words. The hypothesis of the intrinsic evaluation states that similar words should have similar representations.

Thus, to evaluate similarity we first performed a visualization of a sample of words using a T-SNE[7] projection of word embeddings in a two-dimensional space. The experiment was carried out using both the IF and the IAS as weight functions. We examined 50 words, divided into 5 semantic groups. Figure 1 shows how related words are clustered. When we used the IAS as the graph weighting function, the method mostly gathers the elements of the 5 semantic groups. It can be observed that the animals and body parts, are clustered together. On the other hand, 'artifacts', like means of transportation or household appliances, are grouped too. Clothing has its own space, except for *short trousers*, that is isolated. An interesting phenomenon is that *washing machine* is closer to pieces of clothing, and it really could belong to two semantic groups: household appliances and pieces of clothing.

When the IF works as weighting function, the borders of the clustering groups are not so well defined. *washing machine* has still a location close to some pieces of clothe, but this category is spread without order in the two-dimensional space. *Truck* and *frog* are also far from their expected group.

4.1. Similarity and relatedness

In addition, we evaluated the ability of *wan2vec* to capture semantic relations through a similarity task. We used two subsets of the *WordSim-353* [31] benchmark comprised of 353 semantically related term pairs with similarity scores given by humans. This list does not distinguish between the concepts of similarity and relatedness. *Aguirre et al.* [32] split the list into two different sets, one with relatedness scores, containing 252 pairs [EN-WS-353-REL], and another with 203

[7]http://scikit-learn.org/stable/modules/manifold.html#t-sne

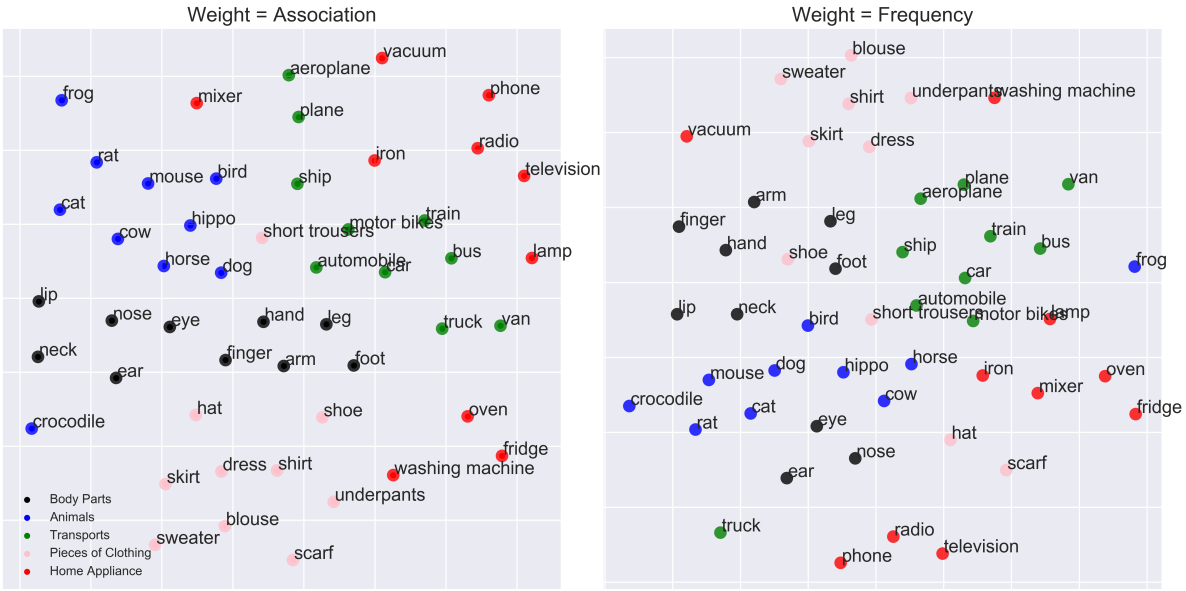

Fig. 1. Projection of word vectors for 5 semantic groups (ten words each). Colors are codified as follows: animals - red, transportation - black, body parts - blue, household appliances - green, clothes - pink.

pairs linked by similarity [EN-WS-353-SIM]. There is an overlap between the lists, all of term pairs belonging to the 353 pairs of *WordSim 353*.

The *WordSim-353* is based on works from the nineties elaborated in the USA. Because of the time and geographical differences, several words from this list are not included in the EAT, a British collection from the early seventies. This fact is not due to a lack of expressivity in WAN, but it is caused because some objects, people and ideas did not exist when the EAT was collected, or they were used in very different contexts. An example can be the word *hardware*, a neologism from the nineties, or *Maradona*, the soccer star from the eighties. As a result, 183 pairs in the list of similarity are in the WAN corpus, while 214 out of 253 in the relatedness compilation. To deal with the non-inclusion of every word of the testing data sets in our EAT WAN, we introduced the concept of overlap in the experiments and calculated the total number of common words between the lists that are being compared. In principle, having large overlaps is a positive feature for *wan2vec*.

We have also tested our method with SimLex-999 [SimLex-999] [33], a resource that contains 999 pairs of words and explicitly quantifies similarity in a way that pairs related by association or relatedness have a low rating. The overlap between the EAT and SimLex-999 is 968, which means that almost every word in the test data is covered by *wan2vec*.

The rest of the data sets we used for our experiments, measure the relatedness of words more than the similarity.

The Amazon Mechanical Turk data [MTurk-287] [34] consists of 287 word pairs, elaborated with the collaboration of the Amazon's Mechanical Turk workers. The overlap with EAT is 203.

MEN [MEN-TR-3k] [35] is an evaluation benchmark that contains 3,000 pairs of randomly selected words. Each pair is scored on a [0, 1] normalized semantic relatedness scale via ratings obtained by crowdsourcing on the Amazon Mechanical Turk. We have a very high overlap with this corpus, 2,727 out of 3,000 word pairs.

The MTURK-771 data set [MTurk-771] [36] evaluates the relatedness between 771 word-pairs. It was obtained with the evaluation of Amazon's Mechanical Turk workers. EAT covers 698 word pairs of this data set.

The RG65 [RG-65][37] collected data for 65 pairs of non-technical words. Their objective was to evaluate the judgments and perceptions about synonymy, and because of these, they used different pairs, in a range from highly synonymous to totally unrelated words. Every word pair in the RG65 is also in the EAT.

Benchmark	***n***	***n (Overlap***	**Wan2vec Dimensions**						
			25	50	100	128	200	300	1000
MC-30	30	30	0.6639	0.8244	0.7817	0.8117	0.7814	**0.8500**	0.7570
WS-353-SIM	204	183	0.6823	0.7637	**0.7839**	0.7481	0.7674	0.7448	0.7276
MTurk-287	287	203	**0.7278**	0.7089	0.6989	0.6834	0.6701	0.6476	0.5626
WS-353-REL	253	214	0.5987	**0.6953**	0.6926	0.6661	0.6855	0.6455	0.58166
MEN-TR-3k	3000	2720	0.7836	0.8000	**0.8031**	0.7947	0.7910	0.7629	0.7089
SimLex-999	999	968	0.4297	0.4548	0.4959	0.5072	**0.5196**	0.5127	0.5052
MTurk-771	771	698	0.6795	0.7083	**0.7315**	0.7170	0.7228	0.7007	0.6674
RG-65	65	65	0.7064	**0.8167**	0.7686	0.8058	0.7605	0.7973	0.7106
Average			0.6590	**0.7215**	0.7195	0.7168	0.7123	0.7077	0.6526

Table 1

Spearman rank order correlations between *wan2vec* predictions and the evaluation benchmarks. The WAN graph was built using the inverted frequency as weighting function.

Finally, we evaluated the *wan2vec* embeddings with the MC-30 [MC-30] [9] benchmark. This list contains 30 word pairs, all of them included in the EAT.

Table 1 reports the Spearman rank order correlations between the previously described datasets and the word embeddings of different dimensions learned on undirected graphs of the WAN EAT, being the weight of the edges given by the IF. Table 2 shows the performance of the *wan2vec* embeddings similarity and relatedness predictions using the IAS as weighting function. For each table, the total number of word pairs (n) and the number of word pair overlap (n (Overlap)) are given.

From these tables it can be seen that, in general, the *wan2vec* embeddings learned on the graph weighted with the IAS weighting function obtained higher correlation with human judgments than the embeddings learned on the graph weighted with the IF function. This was expected because, generally, the association strength is a normalization of frequency that should give a more general idea on the relationship between two words.

It can also be observed that there is a significant overlap between the word pairs of each of the data sets and the vocabulary of our *wan2vec* embeddings. This means that, in general, despite the small size of the resource (only 20,445 nodes) the model has a high level of expressivity, covering most of the words included in the lists.

As for the dimensions of the vectors, 50 and 100 proved to be the most efficient. When using the IF weighting function, the 50 dimensional vectors have a higher average Spearman rank correlation, although the vectors with 100 dimensions obtained the best results on three of the corpora. On the other side, the *wan2vec* embeddings learned on the graph weighted with the IAS weighting function, vectors with 100 dimensions seem to be enough in average and in number of best results.

From the point of view of the benchmarks, the similarities obtained with *wan2vec* achieved correlations of above 0.8 with the MC-30, RG-65, and MEN-TR-3k using both IF and IAS as weighting functions. This is a surprise given that MC-30 and RG-65 are the smallest sets, while MEN-TR-3k is the biggest one. In principle, we expected to have a better performance with the smaller testing sets.

On the other hand, the worst results are achieved with the SimLex-999 resource. This is not because of an small overlap – this is one of the best (96%). Given that the aim of the SimLex data set is to measure only similarity, the words psychologically related have the lowest scores. We believe that this issue affected the performance on this particular resource. In a WAN, words are clearly associated by relatedness. However, this idea is not consistent with the results we obtained with WS-353-SIM and WS-353-REL, where the former has outperformed the latter in every experiment.

In general, the higher scores are achieved with the MC-30. The fact that every word in this list is also in the EAT is not related to the result, because it is also the best when the missing pairs are removed.

4.2. Analysis of the node2vec's Walk Length

We performed an additional analysis which consisted in looking for the optimal walk length of the *node2vec* algorithm. The walk length indicates how deep the algorithm will walk through the graph in order to obtain the node's corresponding embedding. The

Benchmark	***n***	***n (Overlap)***	**Wan2Vec Dimensions**						
			25	50	100	128	200	300	1000
MC-30	30	30	0.6493	0.7725	0.7921	0.8206	0.8480	**0.8582**	0.7409
WS-353-SIM	204	183	0.6956	0.7517	0.7770	0.7633	**0.7775**	0.7623	0.6843
MTurk-287	287	203	**0.7270**	0.7168	0.6976	0.6920	0.6610	0.6661	0.5697
WS-353-REL	253	214	0.6228	0.6967	**0.7051**	0.6854	0.6984	0.6586	0.5711
MEN-TR-3k	3000	2720	0.7825	0.8030	**0.8075**	0.7992	0.7857	0.7646	0.7140
SimLex-999	999	968	0.4206	0.4614	0.4909	0.5009	0.5101	**0.5103**	0.5035
MTurk-771	771	698	0.6769	0.7122	0.7303	**0.7364**	0.7196	0.6973	0.6637
RG-65	65	65	0.7506	**0.8213**	0.7982	0.7850	0.7889	0.7913	0.7153
Average			0.6657	0.7169	**0.7249**	0.7228	0.7230	0.7136	0.6453

Table 2

Spearman rank order correlations between *wan2vec* predictions and the evaluation benchmarks. The WAN graph was built using the inverted frequency as weighting function.

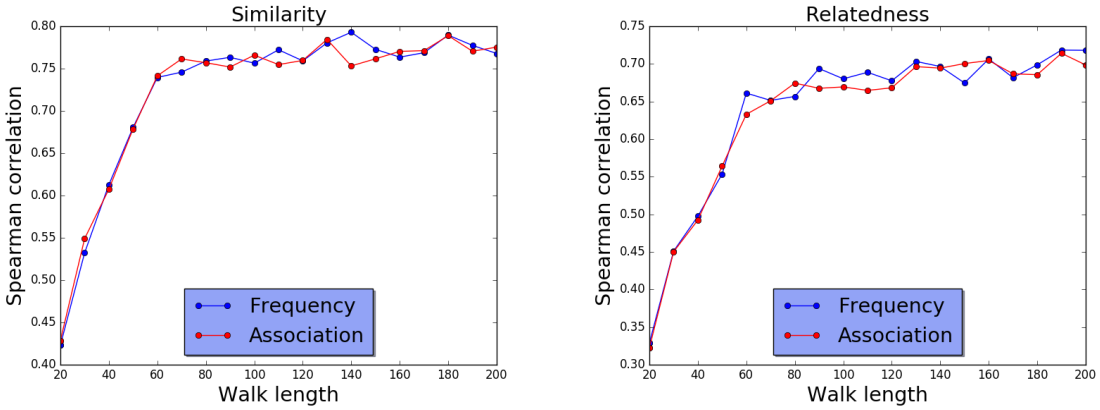

Fig. 2. Spearman rank order correlations obtained with different walk length using both graph weighting functions IF and IAS.

default value is 80, and this is the one we used in the above experiments.

We measured the performance of the *wan2vec* embeddings carrying out the same experiments reported in the previous section, but systematically changing the length of the walk the *node2vec* algorithm traverses for finding the mapping for each node. We evaluated the walk length values from 20 to 200, in intervals of 10. Figure 2 presents the results of the experiments with two data sets the WS-353-REL (Relatedness) and the WS-353-SIM (Similarity).

In every case, the best results are achieved after the walk length of 60, reaching the best performance around 120. However, after this point, the improvement of the quality of the vectors is not remarkable and it can increase the time and complexity for training the model.

4.3. Comparison with Pretrained Vectors

In order to test and compare the quality of our *wan2vec* vectors we also performed the experiments with pretrained vectors, which are presented in Section 4.1. We selected three word embeddings models: *Word2vec*, *GloVe*, and *FastText*. Table 3 presents the general features of the evaluated resources. In all cases the embeddings' dimension is 300 (dimensions), the difference of these three models basically lies in the type and size of the training corpus (Corpus (size)), the total amount of different words (Vocab. size), the training algorithm, and the context windows size. Data about the author of these models is also presented.

Table 4 shows the Spearman rank order correlations between the evaluated pretrained embeddings models and the human judgments (available in the benchmarks). In average, the similarity obtained with the *GloVe* embeddings obtained the worst correla-

Model file	Dimensions	Corpus (size)	Vocab. size	Author	Architecture	Training algorithm	Context window size
Word2vec[8]	300	Google News (100B)	3M	Google	word2vec	negative sampling	BoW - 5
GloVe 6B[9]	300	Wikipedia + Gigaword 5 (6B)	400,000	GloVe	GloVe	AdaGrad	10+10
GloVe 42B[10]	300	Common Crawl (42B)	1.9M	GloVe	GloVe	GloVe	AdaGrad
FastText[11]	300	Wikipedia	Unknown	Facebook	FastText	Skip-gram	Character n-grams [1-5]

Table 3
Description of the pretrained vectors of the three evaluated word embeddings models.

tion scores, whereas the similarity obtained with *FastText* achieved the best performance with an average of 0.7032 of Spearman correlation. However, the performance of the *FastText* embeddings is lower than the best average performance in *wan2vec*, both with the Inverted Frequency weighted graph (0.7215) and with the Inverted Association Strength weighted graph (0.7249).

As with our *wan2vec* embeddings, the highest correlation was achieved with the MC-30 and the RG-65 data sets, obtaining correlations of 0.85 and 0.82, respectively. The pretrained vectors achieved the highest correlation of 0.8119 on the MC-30 with *FastText* embeddings and 0.8171 on the RG-65 data set with *Glove* 42B.

It is also interesting to note that the correlation with the SimLex-999 corpus is the lowest with all three embeddings models, being the best 0.44.

In general, we observed that the *wan2vec* embeddings obtained better correlation with the human judgments than the three pretrained embedding models, which were trained on much larger corpora, thus being more expensive in terms of time and memory.

5. Conclusions and Future Work

In this paper, we introduce *wan2vec* a word embeddings model learned from Word Association Norms instead of large corpora. We applied the algorithm *node2vec* to a graph built over the Edinburgh Associative Thesaurus (EAT), a collection with 23,218 nodes. The result is a set of trained vectors that achieved better correlation with human judgments than other embedding models in the task of similarity and relatedness prediction.

The *Node2vec* algorithm learn embeddings using a random walk that explores the neighborhoods over the nodes to capture a better representation of the graph structure and the diversity of the connectivity.

As for the weight of the edges, we took into account two different functions: inverse frequency (IF) and inverse association strength (IAS). The embeddings learned over the graph weighted with IAS achieved a better average Spearman correlation than the ones learned in the graph with IF weighting function.

We experimented with different walk length distances of *node2vec*, showing that the correlation tends to grow with higher distances and stabilizes above the length of 60. We believe that the default parameter of 80 is fair enough to have representative embeddings since higher values increase the complexity time of the experiments. Finally, we also evaluated a nodes pruning strategy, which consisted in keeping only those nodes that were strongly connected. Following the previous work by *Deyne et al.* [25] we kept only responses that also occur as stimuli and stimuli that were also given as responses. It resulted in a high loss of outcoming nodes and a high reduced overlap, that did not let us give a strong enough comparative benchmark.

The results we are reporting clearly outperform the ones obtained with some well-know pretrained vectors (*word2vec*, *GloVe*, *FasText*) trained in large corpora, like *Wikipedia*. This results are in line with the work by *Deyne et al.* [25], reaffirming the importance of the Word Association Norms as a resource for natural language processing tasks. This kind of resources reflect, to some extent, the mental representation of the words. Additionally, they are valuable for learning distributional representation of words.

There are some simple strategies that would help to improve the quality of the *wan2vec* vectors, for instance, evaluating different types of neighborhoods for

Benchmark	*n*	*n(Overlap)*	GloVe 6B	GloVe 42B	Word2vec	FastText
MC-30	30	30	0.7020	0.7770	0.78860	**0.8119**
WS-353-SIM	204	183	0.6564	0.6912	0.7644	**0.7753**
MTurk-287	287	203	0.7079	0.71851	0.7262	**0.7364**
WS-353-REL	253	214	0.6045	0.6212	0.6324	**0.6886**
MEN-TR-3k	3000	2720	0.7358	0.7338	**0.7691**	0.7649
SimLex-999	999	968	0.3703	0.3736	**0.4413**	0.3765
MTurk-771	771	698	0.6483	**0.6808**	0.6739	0.6719
RG-65	65	65	0.7695	**0.8171**	0.7607	0.7998
Average			0.6493	0.6766	0.6946	**0.7032**

Table 4

Spearman rank order correlations between the evaluated pretrained embeddings models and the human judgments.

the nodes.In the future, we would like to build new *wan2vec* vectors with a larger corpus of cognitive related words. Given that collecting WANs is a hard and time consuming task, our objective is to automatically generate word association norms between pairs retrieved by a medium-size corpus (i.e., BNC), and build a new resource that can account for syntactic, semantic and cognitive connections between words.

References

[1] S.T. Dumais, Latent Semantic Analysis, *Annual Review of Information Science and Technology* **38**(1) (2004), 188–230.

[2] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient Estimation of Word Representations in Vector Space, *Computing Research Repository* **arXiv:1301.3781** (2013). https://arxiv.org/abs/1301.3781.

[3] Z.S. Harris, Distributional Structure, *Word* **10**(2–3) (1954), 146–162.

[4] J. Pennington, R. Socher and C.D. Manning, GloVe: Global Vectors for Word Representation, in: *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. http://www.aclweb.org/anthology/D14-1162.

[5] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, Enriching Word Vectors with Subword Information, *Computing Research Repository* **arXiv:1607.04606** (2016). https://arxiv.org/abs/1607.04606.

[6] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch and A. Joulin, Advances in Pre-Training Distributed Word Representations, in: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, LREC'18, 2018.

[7] C.K. Ogden, *Basic English: A General Introduction with Rules and Grammar*, Paul Treber. http://ogden.basic-english.org/be0.html.

[8] R. Flesch, How Basic Is Basic English?, *Harper's Magazine* **188:1126**, 339–343.

[9] G.A. Miller and W.G. Charlees, Contextual Correlates of Semantic Similarity, *Language and cognitive processes* **6**(1) (1991), 1–28.

[10] S. De Deyne, D.J. Navarro and G. Storms, Associative Strength and Semantic Activation in the Mental Lexicon: Evidence from Continued Word Associations, in: *Proceedings of the 35th Annual Conference of the Cognitive Science Society*, Cognitive Science Society, 2013.

[11] J. Borge-Holthoefer and A. Arenas, Navigating Word Association norms to Extract Semantic Information, in: *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, 2009.

[12] A. Grover and J. Leskovec, node2vec: Scalable Feature Learning for Networks, in: *Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 855–864.

[13] A.M. Collins and E.F. Loftus, A spreading-activation theory of semantic processing., *Psychological review* **82**(6) (1975), 407.

[14] C. H.H., Word Associations and Linguistic Theory, in: *New Horizons in Liguistics*, John Lyons, 1975.

[15] J.F. Sowa, Conceptual graphs as a universal knowledge representation, *Computers & Mathematics with Applications* **23**(2) (1992), 75–93.

[16] J. Aitchison, *Words in the mind: An introduction to the mental lexicon*, John Wiley & Sons, 2012.

[17] G.H. Kent and A.J. Rosanoff, A Study of Association in Insanity, *American Journal of Insanity* **1910**(67) (1910), 317–390.

[18] I. Istifci, Playing with Words: a Study of Word Association Responses, *Journal of International Social Research* **3**(10) (2010).

[19] M. Lafourcade, Making people play for Lexical Acquisition with the JeuxDeMots prototype, in: *SNLP'07: 7th International Symposium on Natural Language Processing*, 2007, pp. 13–15.

[20] G.R. Kiss, C. Armstrong, R. Milroy and J. Piper, *An Associative Thesaurus of English and its Computer Analysis*, Edinburgh University Press, Edinburgh, 1973.

[21] D.L. Nelson, C.L. McEvoy and T.A. Schreiber, *Word Association Rhyme and Word Fragment Norms*, The University of South Florida, 1998.

[22] G. Bel-Enguix, R. Rapp and M. Zock, A Graph-Based Approach for Computing Free Word Associations., in: *Proceedings of the 9th edition of the Language Resources and Evaluation Conference*, LREC'14, 2014, pp. 221–230.

[23] A. Garimella, C. Banea and R. Mihalcea, Demographic-aware word Associations, in: *Proceedings of the 2017 Confer-*

ence on Empirical Methods in Natural Language Processing, EMNLP'17, 2017, pp. 2285–2295.

[24] A. Sinopalnikova and P. Smrz, Word Association Thesaurus as a Resource for extending Semantic Networks., in: *Communications in Computing*, 2004, pp. 267–273.

[25] S. De Deyne, A. Perfors and D.J. Navarro, Predicting Human Similarity Judgments with Distributional Models: The value of word associations., in: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 1861–1870.

[26] D.L. Nelson, V.M. McKinney, N. Gee and G. Janczura, Interpreting the Influence of Implicitly Activated Memories on Recall and Recognition, *Psychological Review* **105**(2) (1998), 299–324.

[27] T. Schnabel, I. Labutov, D. Mimno and T. Joachims, Evaluation Methods for Unsupervised Word Embeddings, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, EMNLP'15, 2015, pp. 298–307.

[28] H. Gómez-Adorno, I. Markov, G. Sidorov, J. Posadas-Durán, M.A. Sanchez-Perez and L. Chanona-Hernandez, Improving Feature Representation based on a Neural Network for Author Profiling in Social Media Texts, *Computational Intelligence and Neuroscience* **2016** (2016), 13.

[29] H. Gómez-Adorno, J.-P. Posadas-Durán, G. Sidorov and D. Pinto, Document Embeddings Learned on Various Types of n-grams for Cross-topic Authorship Attribution, *Computing* (2018), 1–16.

[30] M. Baroni, G. Dinu and G. Kruszewski, Don't count, predict! A Systematic Comparison of context-counting vs. context-predicting Semantic Vectors, in: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1, 2014, pp. 238–247. http://www.aclweb.org/anthology/P14-1023.

[31] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman and E. Ruppin, Placing Search in Context: The Concept Revisited, in: *Proceedings of the 10th International Conference on World Wide Web*, WWW'01, ACM, 2001, pp. 406–414.

[32] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca and A. Soroa, Proceedings of NAACL-HLT 2009, in: *A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches*, 2009.

[33] F. Hill, R. Reichart and A. Korhonen, Simlex-999: Evaluating semantic models with (genuine) similarity estimation, *Computational Linguistics* **41**, 665–695.

[34] 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*, ACM, 2011, pp. 337–346.

[35] E. Bruni, G. Boleda, M. Baroni and N.-K. Tran, Distributional semantics in technicolor, in: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, Association for Computational Linguistics, 2012, pp. 136–145.

[36] G. Halawi, G. Dror, E. Gabrilovich and Y. Koren, Large-scale learning of word relatedness with constraints, in: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2012, pp. 1406–1414.

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