Knowledge Association Network for Measuring Semantic Relatedness

Abstract

Measuring semantic relatedness between two words is a fundamental task for many applications in Natural Language Processing(NLP). Conventional methods mainly utilize the latent semantic hidden in lexical database(WordNet) or text corpus(Wikipedia). They make great achievements based on the distance computing in lexical tree or co-occurrence principle in Wikipedia. However these methods suffer from low coverage and low precision because 1) lexical database contains abundant lexical information but lacks the semantic information; 2) in Wikipedia, two related words may not appear in a window size e.g. synonyms, and two unrelated words may be mentioned together by chance. To compute semantic relatedness more accurately, some other approaches make great efforts based on free association network and get an significant improvement on relatedness measurement. Nevertheless, they need complex preprocessing in Wikipedia. Besides, the fixed score functions they adopt cause the lack of flexibility and expressiveness of model. In this paper, we explore knowledge graph(DBPedia) and wikipedia to construct a knowledge association network which avoids the preprocessing of Wikipedia. And we use distributed vectors instead of fixed score functions to represent the attributes and topological structure of our network respectively. The experiment based on gold dataset shows that our model outperforms the state-of-the-art models.

Introduction

Computing semantic relatedness(SR) between two words is a fundamental task for many applications in Natural Language Processing(NLP) such as lexicon induction(Qadir et al. 2015), Named Entity Disambiguation(Han and Zhao 2010), Keyword Extraction (Zhang, Feng, and Wang 2013) and Information Retrieval(Gurevych, Müller, and Zesch 2007). Besides, in the aspect of spam problem(Sandulescu and Ester 2015) and image classification(Leong and Mihalcea 2011), semantic relatedness measurement plays a great role as well.

Computing semantic relatedness between two words is to get a numerical value which indicates how related two words are. In order to simulate the relatedness judgement of humans, many researchers have worked for semantic relatedness measurement and have made great achievements. In the

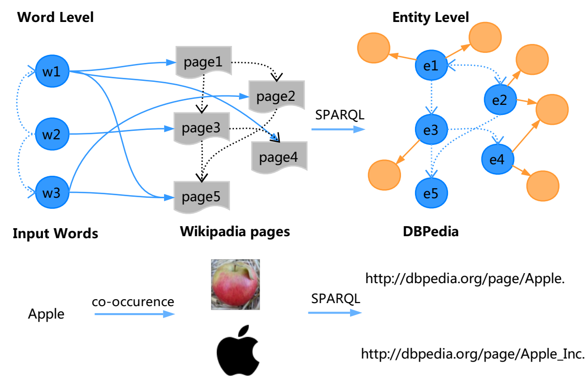

Figure 1: Knowledge Association Network

aspect of data resources for relatedness measurement, there are: i) the methods based on *lexical databases*(Pucher 2007; Zesch, Müller, and Gurevych 2008; Zhu and Iglesias 2017) mainly utilize fixed score functions, such as the path information between two words or the nearest parent common node that two words hold in a lexical tree. These methods play a great part in relatedness measurement and they employ precise and abundant lexical information, but miss the semantic information. ii) *the large corpus* based methods. WikiRelated(Strube and Ponzetto 2006), ESA(Gabrilovich and Markovitch 2007), WLM(Zesch, Müller, and Gurevych 2008) and other methods exploit Wikipedia to compute semantic relatedness following co-occurrence principle, that is, two words are related if they appear in fixed a window size or a sentence collectively. Experimental evaluation shows that these methods outperform the lexical-based methods for the most part. Nevertheless, 1) in some special cases such as synonyms, two related words hardly appear in a sentence. 2) the words that appear in the same sentence may not necessarily be closely related in their semantics but co-occur by chance(Gong, Xu, and Huang 2018).

To overcome the problems of co-occurrence principle, some approaches(Iacobacci, Pilehvar, and Navigli 2015; Pirrò 2012) exploit latent semantic in knowledge graph. Several of them utilize the special knowledge graph BabelNet[1] to annotate different word senses in dumps of Wikipedia, and the others regard that the predicates of triples are important features for words relatedness measurement in knowledge graph. Recently, there are several approaches(Gong, Xu, and Huang 2018; Zhang, Zhu, and Hwang 2015) make great efforts based on the *free association process*[2] in human mind. That is, when given a cue word, the first few words coming into human mind are highly relevant to the cue word. These methods improve the weakness of co-occurrence based methods. However, when comparing the concepts behind words, they mainly adopt some special score functions to judge the different aspects of concepts, such as link information, co-occurrence times, categories of concepts, etc. These fixed functions cause lack of flexibility and are less expressive than the distributed vector representation which is trained to optimize some loss functions. In addition, to get the structured information of concepts, they need significant preprocessing and data transformation efforts in Wikipedia.

In this paper, we use DBPedia and Wikipedia jointly where an entity corresponds to a title of Wikipedia page. In this way we avoid the preprocessing of information extraction in Wikipedia, as DBPedia contains structured information extracted from Wikipedia. Figure 1 shows that, for a given word(*apple*), there are some Wikipedia pages that contain this word(*apple*), and the titles of pages correspond to the entities in DBPedia. These entities and their relationships constitute the entities network, where i) some relationships play a role of attributes(orange circles in figure 1); ii) some others just connect two entities without explicit semantic, such as two entities appear in the same Wikipedia page(blue circles), which form the topological structure of entity network. To get an approximation to human relatedness judgement more accurately, we construct a knowledge association network to capture the relatedness of word-to-word, word-to-entity and entity-to-entity. The contributions made in this paper include:

1. Based on the free association network, we propose a knowledge association network that is built on DBPedia which avoids the preprocessing in Wikipedia and still keeps the complete semantics in our network.

2. For the relatedness measurement of entity-to-entity in our model, we use two different unsupervised methods to represent the attributes and the topological structure of entity network rather than fixed score functions. Finally we combine the relatedness of word-to-word, word-to-entity, entity-to-entity methodically.

3. We conduct experiments on standard dataset of semantic relatedness, and the result shows that our model gets higher correlation coefficient than the state-of-the-art models.

This paper is organized as follows. We give the definition and construction of knowledge association network firstly. Then we elaborate the knowledge association network model to compute relatedness scores. Finally, we display detailed illustrations of experiment results.

[1]http://babelnet.org

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

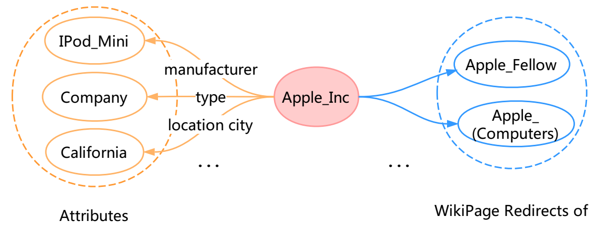

Figure 2: The semantics around an entity

Knowledge Association Network

It has long been thought that when humans measure the relatedness between a pair of words, a deeper reasoning which requires a large amount of knowledge is triggered to compare the concepts behind the words. There are many data resources that contain concepts which are associate with words such as Wikipedia, Wordnet and DBPedia etc. Wordnet provides precise lexical information but lacks adequate semantic information. Wikipedia is a large corpus where a page describes a concept. DBPedia contains abundant structured knowledge consisting of a great number of facts which are extracted from wikipedia.

Inspired by the free association network in semantic relatedness between two words(Gong, Xu, and Huang 2018; Zhang, Zhu, and Hwang 2015), we consider the DBPedia as concepts database to avoid the significant preprocessing and data transformation efforts in wikipedia. To measure the relatedness between two entities, we consider two major factors in DBPedia: attributes information and topological structure. The attributes of an entity include the properties, categories, ontology information and some other information which enhances the entity itself. Topological structure reflects the relations between other entities on the basis of a special predicate in DBPedia: *WikiPageRedirectOf*. That means if two entities are connected by this predicate, they appear in the same Wikipedia page.

There is an example in figure 2, for the technology company Apple which is described as *Apple Inc* in DBPedia, we get its attributes, that is, "Apple is the manufacturer of IPod Mini(properties)"; "Apple is a company(categories)" and etc. And the relationship descriptions(*predicates in triples)* are named on the basis of ontology language that contains affluent semantics. In the aspect of links among other co-occurrence entities, there are entities *Apple_Fellow* and *Apple_(Company)* in accordance with the special relationship *WikipageRedirectOf*.

DBPedia contains natural attributes network structure and abundant semantics. We define a set of entities $E_w = \{e_1, e_2, ..., e_i\}$ that represent the semantics behind a word and e_i is an entity in DBPedia. We refer to the attributes of an entity as an attributes graph $G_{attr} = \{a_1, a_2, ..., a_j\}$ and define topological structure as $G_t = G(E, R_{redirect})$, where a_i denotes an attribute and E is a set of entities, $R_{redirect}$ is the edge set formed by *WikiPageRedirectOf*.

Definition *Knowledge association network can be modelled as a graph $G = (W, E, R)$ where w is the word set in vocabulary, E is the entity set contained in DBPedia, and edge set R denotes the relationships including word-to-word(R_w), entity-to-entity(R_e), and word-to-entity(R_{we}).*

Network Construction A necessary work in network construction is to build the mapping between words and concepts(entities). This comes in handy in Wikipedia pages. The mapping between words and Wikipedia page reflects the structure of association network naturally, and fortunately there is a 1-to-1 match between entity in DBPedia and page in Wikipedia. In other words, Wikipedia page and it's corresponding DBPedia entity elaborate the same concept. Consequently we can construct knowledge association network based on this natural mapping.

There is a special attribute called *WikipageID* that reflects the mapping between entities in DBPedia and pages in Wikipedia by the unique id. The id can be obtained by the *Gensim*[3] that is a free Python library designed to automatically extract semantic topics from documents. By the aid of unique page id, we get the unique corresponding entity in DBPedia by SPARQL endpoint[4]. For example, the id of Wikipedia page *Apple Inc* is 856, we can use a simple query to get the unique entity name *Apple_Inc*:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?E WHERE {
    ?E dbo:wikiPageID 856.
}
```

Semantic Relatedness Measurement

We give an overview of our model in figure 3 where solid lines lead the flow of model and dotted lines demonstrate that there is an additional function from source part to target part. This figure illustrates the construction of our network and the relatedness measurement. For the words in vocabulary, we can get a mapping between words and concepts(pages in Wikipedia) by corpus statistics. Then we query the unique entity by the page id in DBPedia SPARQL endpoint. For the layer of entity-to-entity, we divide it into attributes and topological structure and embed them by different models. As for the semantic relatedness computing, we consider three layers relatedness measurement: word-to-word, word-to-entity and entity-to-entity when it comes to knowledge association network.

word-to-word

The semantic relatedness in word layer is mainly measured by 1) distributed vector representation such as word2vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014) etc. 2) word co-occurrence, which means two words are relevant if they appear in a window size K. Experimental results prove that distributed vector representation works better in computing semantic relatedness(Mikolov et al. 2013). Therefore in this paper, we abandon co-occurrence-based methods and adopt word2vec to train the Wikipedia corpus to product effective vector representation for each word. Let $\overrightarrow{v}_i$ and $\overrightarrow{v}_j$ denote the vector representation of w_i and w_i, and $f(w_i, w_j)$ is the semantic relatedness score. The vector representation can be utilized to calculate the semantic relatedness based on cosine function. Formally, we get:

$$f_w(w_i, w_j) = cos(\overrightarrow{v}_i, \overrightarrow{v}_j) = \frac{\overrightarrow{v}_i \cdot \overrightarrow{v}_j}{\|\overrightarrow{v}_i\| \, \|\overrightarrow{v}_j\|} \qquad (1)$$

The word2vec algorithms include skip-gram and CBOW models, using either hierarchical softmax or negative sampling. The combination of skip-gram and negative sampling are used frequently and effective experimentally. We choose this training program accordingly. The detailed parameter setting can be seen in the part of experiment.

word-to-entity

In the knowledge association network, there is a one-to-many relationship between words and entities in DBPedia. For a given word, several relevant entities will rise from our network due to the word ambiguity. To measure the degree of association between word(w) and entity(e), 1) some researchers(Pirrò 2012) take the co-occurrence times between w and c as the judgement of relatedness, which is insensitive for some common words like *this*, *that* and so on. 2) and some other related works(Gong, Xu, and Huang 2018) consider w and e are closely related if e is the only semantic meaning for word w. They compute the degree of strong connections between only anchor words and their out link concepts based on the follow equation:

$$LP(w, c) = \sum_{P} \sum_{w \in S} \frac{\sum_{w' \in S} tf\_idf(w^{'}, c)}{\sum_{c^{'} \in c(w)} \sum_{w^{'} \in S} tf\_idf(w', c')} \qquad (2)$$

where P indicates a wiki page, S represents one sentence in a wikipage that contains the word w, and $w^{'}$ means every contextual word in S. $c(w)$ is a set of concepts which are linked from anchor word w. This method just considers anchor words and out link concepts, but ignores the relationship between current page and the anchor texts.

It is well known that tf-idf is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. From our point of view, in order to measure the relatedness between words and entities, we adopt tf-idf measurement. Specially, we evaluate the relationship of mapping from words to Wikipedia pages as:

$$f_{we}(w_i, e_j) = \frac{tf\_idf(w_i, e_j)}{\sum_{e^{'} \in e(w)} tf\_idf(w_i, e^{'})} \qquad (3)$$

where $e(w)$ is the entities set that are associated with w, and the wikipage of $e^{'}$ contains the word w.

[3]https://radimrehurek.com/gensim/wiki.html

[4]http://dbpedia.org/sparql

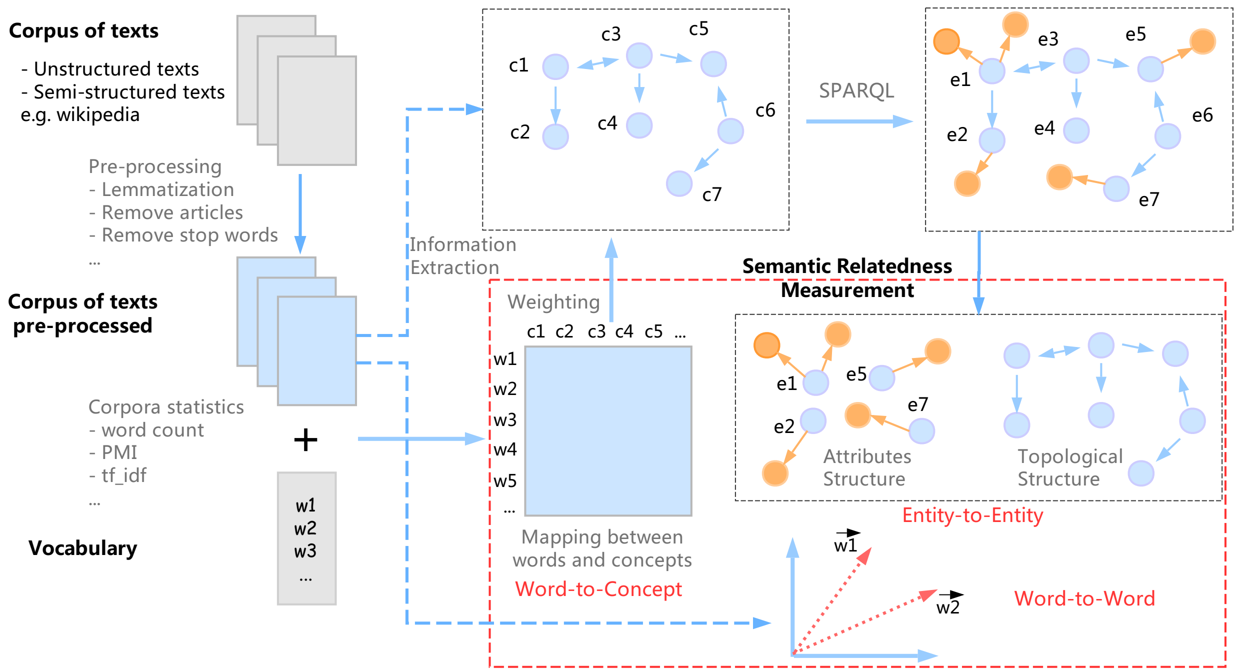

Figure 3: Association Network in Semantic measurement

entity-to-entity

The knowledge association network in entity-to-entity level is fundamentally a multi-relational graph where an entity is described by discrete attributes and topological structure collectively. It is unreasonable to just consider either of the these information. Two entities may hold totally different attributes but they appear in the same Wikipedia page and vice verse. The part of attributes holds the detailed semantic information such as person A is the friend of B, person B is the member of organization C etc. The topological structure in entity-level reflects the co-occurrence of entities. Things get quite different in representation of semantic in these two structure. Consequently, we adopt two different methods to obtain vector representation of the attributes and the topological structure.

embedding for attributes The straightforward method to embed a set of attributes around an entity is *one-hot*. Nevertheless, a surprisingly high number of attributes in DBPedia brings a insoluble problem for *one-hot* because of the excessive dimensions. From our point of view, there exists a *1-to-1* relationship between entities and their attributes, which is interpreted as a translation operation on the low-dimension embeddings of entities(Bordes et al. 2013; Wu et al. 2017). Suppose there are N different attributes in our network and the attributes space is denoted as $\mathbb{R}^{|N|\times|d|}$, where d the dimension of embedding for one attribute. Inspired by the translation embeddings on knowledge graph, we combine the relationships and entities to minimize a margin ranking loss over the attributes graph G_{attr}:

$$\mathcal{L} = \sum_{(a,b)\in G^+_{attr}} \sum_{b^-\in G^-_{attr}} [\ell + cos(a,b) - cos(a,b^-)]_+$$

where $[x]_+ = max(0,x)$, and ℓ is a margin hyperparameter.

In our model, the input data G_{attr} is a set of (h,r,t) triples, consisting of a head entity h, a relation r and a tail entity t. The positive entity pair (a,b) is sampled from attributes network G_{attr}, and we select uniformly at random to get positive sample G^+_{attr} in two strategies: (i) a consists of the bag of h and r, while b consists of only t; (ii) a consists of h, b consists of r and t. Negative entities b^- are sampled from the set of possible triples G^-_{attr}. We utilizes a k-negative sampling strategy(Mikolov et al. 2013) to select k negative pairs for each batch update. The optimization of method inherits the strategy of stochastic gradient descent(SGD). Each SGD step is one sampling from G^+_{attr} in the outer sum.

As a result, we take the entities attributes graph G_{attr} of (h,r,t) triples as inputs to train the model. For each entity and relation in graph G_{attr}, there is a fixed-length vector which can then be used to compute semantic relatedness via cosine function.

embedding for topological structure The connections among a great deal of entities include many semantic relations such as *Apple_Inc* is the *manufacturer* of *IPod*, person

A is the *wife* of *person B* etc. Most of these relations contain specific semantic knowledge, but there is a relation named *WikiPageRedirectOf* which connects two entities if one entity's anchor text description is mentioned in the corresponding Wikipedia page of the other. We can get the topological structure G_t among entities on account of this redirection relation. When somebody browses a wikipage and he wants to check a few out links which are contained in current page, the most related out link will appear in his mind firstly. This situation indicates that the edges in G_t have different transition probability, which is a probability graph model shown in figure 4.

The most straightforward way to weight the edges is to consider the occurrence number of the anchor text in a wikipage. We intergrate the anchor text as a single term t_i for an entity e_i. Let $cnt(e_i, e_j)$ denote the co-occurrence frequency of the term t_j of e_j that appears in page of e_i. Formally, for an entity-to-entity weighted edge $< e_i, e_j >$, we have:

$$W_{cnt}(e_i, e_j) = \frac{cnt(e_i, e_j)}{\sum_{e' \in p_i} cnt(e_i, e')}$$

where $W_{cnt}(e_i, e_j)$ denotes the probability from e_i to e_j. p_i is the corresponding Wikipedia page of e_i, and contains some anchor texts for each $e^{'}$ in p_i. However, just consider anchor text frequency would give some general frequent terms high degree of relatedness. Thus we adopt tf_idf instead of co-occurrence principle to weigh how important an entity(anchor text) is to another(current page). There is:

$$W_{tf\_idf}(e_i, e_j) = \frac{tf\_idf(e_i, e_j)}{\sum_{e' \in p_i} tf\_idf(e_i, e')}$$

Then we can get the transition probability from e_i to e_j:

$$\begin{cases} \text{P}(e_j|e_i) = W(e_i, e_j) \\ \text{W} \in (W_{cnt}, W_{tf\_idf}) \end{cases}$$

In consideration of topological structure, in network, some embedding methods(Perozzi, Al-Rfou, and Skiena 2014; Grover and Leskovec 2016) for nodes are mainly inspired by Skip-gram model. They represent a network as a "document". The same way as a document is an ordered sequence of words, one could sample sequences of nodes from the underlying network and turn a network into an ordered sequence of nodes. In this paper, to get expressive vector representations for nodes, we follow Skip-gram model to get network embedding. Our first priority is to determine the sampling strategy. One excellent work called *node2vec*(Grover and Leskovec 2016) proposes a flexible neighborhood sampling strategy which allows us to smoothly interpolate between BFS and DFS. They propose a search bias we called a:

$$\text{a}_{pq}(t, x) = \begin{cases} 1/p & if\ d_{tx} = 0 \\ 1 & if\ d_{tx} = 1 \\ 1/q & if\ d_{tx} = 2 \end{cases} \tag{4}$$

It can be exhibited in the right of figure 4, edge labels indicate search biases a. Suppose we start a random walk which traverses the edge (t, v), now we are in node v and next can walk to (t, x_1, x_2). There are $d_{tt} = 0$, $d_{tx_1} = 2$ and $d_{tx_2} = 2$, so we can get the result biases shown in figure 4. Note that if node t and x_2 are connected, bias from v to x_2 would be 1.

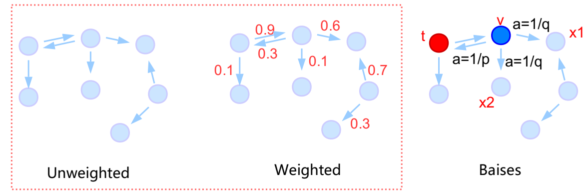

Figure 4: Unweighted, Weighted topological structure and the random walk bias

The node embedding is a maximum likelihood optimization problem following Skip-gram model which uses one word to predict its context. Analogously, given a sequence of entities $(e_0, e_1, ..., e_i, ...e_l)$ sampled from the network G_t, for an entity e_i, we utilize it to predict the neighborhood entities, which is to estimate the likelihood:

$$Pr = ((e_0, e_1, ..., e_{i-1}, e_{i+1}, ..., e_l)|e_i) \tag{5}$$

We define $N(e_i)$ as the context entities around the entity e_i, and we introduce a mapping function $\Phi : e \in E \to \mathbb{R}^{|E| \times d}$ where E is not the edge set of G_t but the entity set i.e. node set. Φ is represented as a $|E| \times d$ matrix of parameters which will be trained to get. For each $e_i \in E$, we will get a d-dimension vector. There is the loss function to minimize:

$$min -logPr(N(e_i)|\Phi(e_i)) = -log \prod_{e^{'} \in N(e_i)} Pr(e^{'}|\Phi(e_i)) \tag{6}$$

For $e^{'} \in N(e_i)$, it have a symmetric effect with e_i in feature space(Grover and Leskovec 2016). So we adopt the softmax function to normalize the likelihood probability:

$$Pr(e^{'}|\Phi(e_i)) = \frac{exp(\Phi(e^{'}) \cdot \Phi(e_i))}{\sum_{e_k \in E} exp(\Phi(e_k) \cdot \Phi(e_i))} \tag{7}$$

Finally, We optimize likelihood function (6) using SGD.

measurement We can get the embedding for an entity e_i, that consists of attributes information embedding($\overrightarrow{va}_i$) and topological structure embedding($\overrightarrow{vt}_i$). Formally, we refer to the relatedness of entity-to-entity as:

$$f_e(e_i, e_j) = \alpha cos(\overrightarrow{va}_i, \overrightarrow{va}_j) + (1 - \alpha) cos(\overrightarrow{vt}_i, \overrightarrow{vt}_j) \tag{8}$$

where $\alpha \in [0, 1]$ is to adjust the weights of two parts.

Word Semantic Relatedness *F*

We consider three layers for final semantic relatedness measurement including word-to-word, word-to-entity and entity-to-entity. The word-to-entity and entity-to-entity are combined as concept-layer relatedness $F_c(w_i, w_j)$, and we refer to word-to-word to word-layer relatedness named $F_w(w_i, w_j)$. Formally, We have:

$$F_c(w_i, w_j) = \sum_{e_i \in E_i} \sum_{e_j \in E_j} f_{we}(w_i, e_i) f_e(e_i, e_j) f_{we}(w_j, e_j) \tag{9}$$

in which E_i is the entities set that is associated with word w_i. The final word semantic relatedness measurement are:

$$F(w_i, w_j) = \varphi F_w(w_i, w_j) + (1 - \varphi) F_c(w_i, w_j) \tag{10}$$

$\varphi \in [0, 1]$ trades off the weight of F_w against F_c.

Experiment

In this section, we conduct extensive experiments on different datasets which contain the semantic measurement by human perceptions. We compute the Pearson correlation coefficient γ, Spearman correlation coefficient ρ and harmonic mean coefficient $\mu = \frac{2\gamma\rho}{\gamma+\rho}$ between results of experiments and scores of human judgement to evaluate the performance of our model.

Dataset

The knowledge association network KAN_{wiki} is conducted based on the Wikipedia[5] and DBPedia[6]. The details about the basic dataset are shown in table 2. You will find the number of entities is larger than documents in Wikipedia, since the entities set contain entities extracted not only from Wikipedia but also some other semantic dataset such as ontololy language, YAGO and so on. It is necessary to preprocess the Wikipedia before constructing network KAN_{wiki}. As shown in figure 3, for each page in Wikipedia, we remove the stop words and punctuation, ignore the shorter pages whose words number less than 50, and ignore some special namespaces such as *Category, File, Template* and so on.

	Documents	Date
Wikipedia	5.5M	2016-10
DBPedia	6.6M	2016-10
	Entities	Date

Table 1: Wikipedia and DBpedia Information

Evaluation

A great number of datasets record the scores of human quantitative judgement for semantic relatedness. We evaluate KAN_{wiki} on three frequently used datasets that are listed in table 2. Based on the standard dataset, we compare our model to other existing models, such as ESA(Gabrilovich and Markovitch 2007), SSA(Hassan and Mihalcea 2011), word2vec(Mikolov et al. 2013), SaSA(Wu and Giles 2015), AN(Zhang, Zhu, and Hwang 2015) and HAN(Gong, Xu, and Huang 2018).

[5]https://dumps.wikimedia.your.org/

[6]https://wiki.dbpedia.org/downloads-2016-10

dataset	word pairs	score scope	reference
MC	30	[0,4]	Miller&Charles1991
RG	65	[0,4]	Rubenstein&Goodenough1965
WS353	353	[0,10]	Finkelstein et al.2002

Table 2: Wikipedia and DBpedia Information

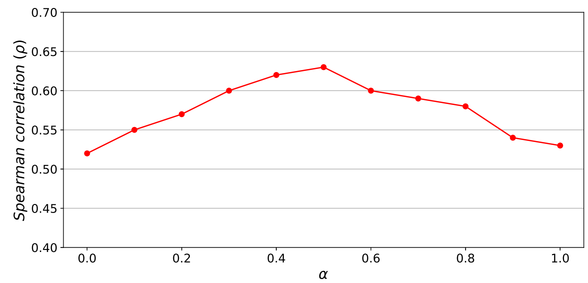

Figure 5: α tuning on WS-Rel only consider entity-to-entity layer

Parameters tuning In this paper, it is necessary to determine the following parameters:

- Recall word-to-word, we train word2vec in Wikipedia to get the vector representations for words. And we adopt *100 dimension, 30 window size, Skip-gram model and negative sampling* for word2vec following *KAN*.
- In the section of attributes embedding, we set *margin* $\ell = 0.05$, *dimentsion* $d = 100$, *negative sampling number* $k = 50$ for attributes graph embedding, and we set the learning rate of SGD as 0.1 to optimize the margin ranking loss.
- In the section of embedding for topological structure, the parameters p and q control the search bias a. The Skip-gram model is used for training the sequences of random walk, and we set the *100 dimension, 10 window size* as the basic parameters in Skip-gram model.
- α is proposed for the balance of attributes information and topological structure. φ trades off the weight of word-layer against concept-layer.

In order to get the optimal correlation, we pick *WS-Rel*(Agirre et al. 2009) to tune the p, q and α. since there are not many comparison systems in the literature that report results on this dataset. This dataset contains 252 pairs of words along with relatedness judgement. We compute word semantic relatedness just on entity-to-entity layer(f_e) and find the optimal values $p = 3$ and $q = 1$ respectively. As for α, as shown in figure 5, Spearman correlation(ρ) increases evidently when the importance of topological structure is raised. And we get the optimal values for α to be 0.5, which means attributes information and topological structure play the same role for semantic relatedness measurement.

Another parameter φ trades off the weight of word-layer relatedness F_w against concept-layer relatedness F_c. We set tuning rate of φ is 0.1. Figure 6 shows the results w.r.t the

Model	λ			ρ			μ		
	MC	**RG**	**MS353**	**MC**	**RG**	**MS353**	**MC**	**RG**	**MS353**
ESA	0.588	- -	0.503	0.727	- -	0.748	0.650	- -	0.602
SSA	0.879	0.861	0.590	0.843	0.833	0.604	0.861	0.847	0.597
word2vec	0.852	0.834	0.633	0.836	0.812	0.645	0.844	0.823	0.639
SaSA	0.886	0.882	0.733	0.855	0.851	0.739	0.870	0.866	0.736
AN_{wiki}	0.865	0.858	0.740	0.848	0.843	0.813	0.856	0.850	0.775
HAN_{wiki}	0.886	0.884	0.772	0.860	0.857	0.826	0.873	0.870	0.798
KAN_{cnt}	0.850	0.826	0.630	0.836	0.805	0.633	0.842	0.816	0.631
KAN_{tf_idf}	**0.892**	**0.887**	**0.789**	**0.872**	**0.865**	**0.841**	**0.882**	**0.876**	**0.814**

Table 3: Pearson-λ, Spearman-ρ, harmonic mean-μ on the word relatedness datasets

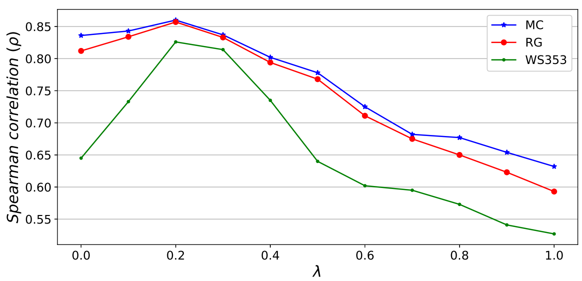

Figure 6: Performance with value of λ

multiple value of φ on KAN_{tf_idf}, and when $\varphi = 0.2$, we get the largest Spearman correlation(ρ). Obviously, F_w have a leading role and our F_c make a great supplement for final semantic relatedness measurement.

Comparions results Evaluation result of word semantic relatedness on different correlation coefficient is shown in table 3. Recall embedding for topological structure of our network, there are two strategies to weight the relationship among entities: 1) $W_{cnt}(e_i, e_j)$ denote the co-occurrence frequence of e_j in page of e_i; 2) $W_{tf\_idf}(e_i, e_j)$ adopt tf_idf to judge how import an entity is to the other. Based on these two weight strategies, we construct KAN_{cnt} and KAN_{tf_idf} respectively. We can see that the KAN_{tf_idf} outperforms KAN_{cnt} in different datasets and measurement coefficients, since tf_idf increases proportionally the number of times a term(t) appears in the page of an entity. And the value of tf-idf is offset by the number of pages in Wikipedia that contain the item t, which helps to adjust the weight for the fact that some items appear more frequently in general.

When compared with other methods shown in table 3, our method performs better. AN_{wiki} and KAN_{wiki} get excellent performance on word semantic relatedness on the idea of *free association network*, which improve the weakness of co-occurrence-based methods. KAN_{tf_idf} measure semantic relatedness among concepts on the shoulder of AN_{wiki} and KAN_{wiki}. We adopt two different model to capture the semantic of attributes(G_{attr}) and topological structure(G_t) in KAN_{tf_idf} and make the model more flexible and expressive.

Conclusion

In this work, we focus on computing semantic relatedness to get an approximation to human judgement. We utilize the Knowledge Graph DBPedia which is derived from Wikipedia as background knowledge to construct a knowledge association network. To measure the word semantic relatedness, we consider three kinds of relationships which consist of word-to-word, word-to-entity and entity-to-entity, and we adopt two difference models to capture the semantic of attributes and topology structure instead of fixed score functions in our network. The experiments based on golden dataset show that our model outperforms the state-of-the-art models in semantic relatedness measurement.

References

Agirre, E.; Alfonseca, E.; Hall, K. B.; Kravalova, J.; Pasca, M.; and Soroa, A. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA*, 19–27.

Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, 2787–2795.

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 1606–1611.

Gong, X.; Xu, H.; and Huang, L. 2018. HAN: hierarchical association network for computing semantic relatedness. In

Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 855–864.

Gurevych, I.; Müller, C.; and Zesch, T. 2007. What to be? - electronic career guidance based on semantic relatedness. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Han, X., and Zhao, J. 2010. Structural semantic relatedness: A knowledge-based method to named entity disambiguation. In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, 50–59.

Hassan, S., and Mihalcea, R. 2011. Semantic relatedness using salient semantic analysis. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, 95–105.

Leong, C. W., and Mihalcea, R. 2011. Measuring the semantic relatedness between words and images. In *Proceedings of the Ninth International Conference on Computational Semantics, IWCS 2011, January 12-14, 2011, Oxford, UK*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1532–1543.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 701–710. ACM.

Pirrò, G. 2012. Reword: Semantic relatedness in the web of data. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*.

Pucher, M. 2007. Wordnet-based semantic relatedness measures in automatic speech recognition for meetings. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Qadir, A.; Mendes, P. N.; Gruhl, D.; and Lewis, N. 2015. Semantic lexicon induction from twitter with pattern relatedness and flexible term length. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 2432–2439.

Sandulescu, V., and Ester, M. 2015. Detecting singleton review spammers using semantic similarity. In *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, 971–976.

Strube, M., and Ponzetto, S. P. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, 1419–1424.

Wu, Z., and Giles, C. L. 2015. Sense-aaware semantic analysis: A multi-prototype word representation model using wikipedia. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 2188–2194.

Wu, L.; Fisch, A.; Chopra, S.; Adams, K.; Bordes, A.; and Weston, J. 2017. Starspace: Embed all the things! *CoRR* abs/1709.03856.

Zesch, T.; Müller, C.; and Gurevych, I. 2008. Using wiktionary for computing semantic relatedness. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, 861–866.

Zhang, W.; Feng, W.; and Wang, J. 2013. Integrating semantic relatedness and words' intrinsic features for keyword extraction. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2225–2231.

Zhang, K.; Zhu, K. Q.; and Hwang, S. 2015. An association network for computing semantic relatedness. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, 593–600.

Zhu, G., and Iglesias, C. A. 2017. Computing semantic similarity of concepts in knowledge graphs. *IEEE Trans. Knowl. Data Eng.* 29(1):72–85.