An Association Network for Computing Semantic Relatedness

Keyang Zhang [1] and **Kenny Q. Zhu** [2]
Shanghai Jiao Tong University, Shanghai, China
[1]keyangzh@gmail.com, [2]kzhu@cs.sjtu.edu.cn

Seung-won Hwang
POSTECH, Pohang, Republic of Korea
swhwang@postech.ac.kr

Abstract

To judge how much a pair of words (or texts) are semantically related is a cognitive process. However, previous algorithms for computing semantic relatedness are largely based on co-occurrences within textual windows, and do not actively leverage cognitive human perceptions of relatedness. To bridge this perceptional gap, we propose to utilize free association as signals to capture such human perceptions. However, free association, being manually evaluated, has limited lexical coverage and is inherently sparse. We propose to expand lexical coverage and overcome sparseness by constructing an association network of terms and concepts that combines signals from free association norms and five types of co-occurrences extracted from the rich structures of Wikipedia. Our evaluation results validate that simple algorithms on this network give competitive results in computing semantic relatedness between words and between short texts.

1 Introduction

Computing semantic relatedness between two words (or texts) is a fundamental task in natural language processing, artificial intelligence and information retrieval. Strictly speaking, semantic relatedness is a more general notion than semantic similarity as it captures not only closeness between two objects within a type hierarchy (e.g., river and stream), but also any other relations (e.g., river and boat) (Budanitsky and Hirst 2006). Traditionally, semantic similarity has been computed either within some lexicon (Jarmasz 2003; Resnik 1995; Jiang and Conrath 1997; Lin 1998) or by comparing the distributional properties of contexts (Deerwester et al. 1990; Gabrilovich and Markovitch 2007; Hassan and Mihalcea 2011). On the other hand, semantic relatedness has been largely modeled by co-occurrences within a window in a large text corpus.

For both similarity and relatedness, co-occurrences play a central role, hence how they are extracted and combined can significantly influence the quality of relatedness computation. So far, dozens of similarity functions (McGill 1979) have been proposed for IR, all of which involving co-occurrences in one way or another, but few achieve satisfactory results on both similarity and relatedness. The reason for such limited success, we argue, is that, since similarity and relatedness are ultimately human perceptions and thus evaluated against human annotated scores, simple window-based co-occurrences, often contaminated with noises, offer insufficient signals to match the human perception. In other words, there exists a perceptional gap between the relatedness perceived by humans and the co-occurrences we collect from text corpora.

As a human perception signal to bridge such gap, we consider a well-studied psychological process called *free association*. In free association, a person is given a *cue* word and is asked to produce the first word that comes to her mind as the *response*. Previously, a number of free association experiments by psychologists resulted in a few data sets called *free association norms*. Table 1 is a fragment of the free association norms collected by University of South Florida (Nelson, McEvoy, and Schreiber 2004), known as Florida Norms from now on.

Table 1: The strongest reponses to the cue word "river"

Cue	Response	Strength
river	lake	15/150
river	stream	15/150
river	water	9/150
river	flow	8/150
river	boat	7/150
river	canoe	7/150

Each row of the data contains a cue word, a response word and the strength of the association (a fraction of the people who responded with this pair of association in the experiment). The free association norms can be viewed as a network in which nodes are the words, and edges carry the strengths. One can see that edges in this network connect both similar pairs (e.g., river and stream) and related ones (e.g., river and boat), which seems ideal for computing semantic relatedness. However, this network suffers from two limitations. First, the number of cue words in these datasets ranges from 100 to 5000, which means only the most common English words are covered and the scale of such a network is too small for predicting the relatedness score between two arbitrary words. Second, due to the cost of free association experiments, the number of human subjects is usually small. A cue word is typically presented to a few

dozens to 1000 subjects, yielding a few dozens unique responses. Thus the free association network is fairly sparse.

In this paper, we propose a novel approach to construct a large-scale, comprehensive association network of English terms and concepts by combining semantic signals from both free association norms and Wikipedia. Wikipedia is a large, high-quality text corpus from which co-occurrences can be drawn. In the past, people primarily extracted co-occurrences between terms within the Wikipedia article body. Instead, we leverage the rich structure within Wikpedia, to extract 5 types of co-occurrences, which are then aggregated into a single, universal association strength score by learning from the strengths of the free association norms. Such scores are used to weight the edges in the proposed association network. This network can be thought of as an expanded, smoothed version of the free associate network, and can be used to simulate how an average human being associates one concept to another in her mind. We would then use this association network to compute the semantic relatedness between terms and short texts.[1]

In summary, this paper makes three main contributions.

1. We extract 5 different types of co-occurrences from Wikipedia and construct a "synthetic" association network by training on free association norms (Section 2);
2. We empirically show that free association is a competent alternative source of knowledge for computing semantic relatedness, and our "synthetic" association network effectively simulates free association and resolves its limitations (Section 3.2 and Section 3.3);
3. We propose algorithms to compute semantic relatedness based on the constructed association network , which outperform state-of-the-art methods (Section 3.4).

2 Our Approach

In this section, we first define the proposed *association network*, then show how to populate such network. We then propose algorithms to compute relatedness using this network, and finally conclude with some discussions.

2.1 Association network

A *super node* s represents a set of synonymous terms and their corresponding Wikipedia concepts (or article pages), denoted as (T, C), where T is a set of terms and C is a set of Wikipedia concepts. For example, $(\{apple, apples\}, \{Apple, Apple\ Inc.\})$ is one such super node. Given a term t, we can generate a super node s by Algorithm 1. $def_c(t)$ returns a set of Wikipedia concepts defining t, while $def\ (c)$ returns a set of terms defined by c. We say t is defined by c if t, as an anchor text, links to c at least 10% of the time, and c is being linked from t at least 10% of the time. We found the results to be insensitive to the value of 10%, which was empirically determined.

Our association network is a weighted directed graph $G(V, E)$, with $w(e)$ denoting the weight of edge e ($e \in E$). Each vertex in the graph is a super node s, and an edge $e(u, v)$ ($u, v \in V$) indicates u can associate to v, with strength $w(e)$. For all $u \in V$, strength is normalized:

$$\sum w(u, v) = 1 \quad (1)$$

Algorithm 1 Generate super node

```
1: function BOOTSTRAP(term t)
2:     T ← {t}, C ← {}
3:     while T or C is updated do
4:         for t ∈ T do
5:             C ← C ∪ def_c(t)
6:         for c ∈ C do
7:             T ← T ∪ def (c)
8:     return (T, C)
```

2.2 Network construction

Given a set of terms T_0, we populate an association network $G(V, E)$ in two steps: first determine the vertex set V in our network, and then determine edge set E and estimate the association strengthes for edges in the network.

To determine the vertex set V of our association network G, we run Algorithm 1 for every $t \in T_0$, such that the set of all output super nodes is V. Algorithm 1 ensures V to have the following property [2]:

Lemma 1 *Each term t in T_0 appears in exactly one vertex of G, and no two vertices share an identical concept c.*

To determine the edge set E of our association network and the association strength of each $e \in E$, we tap into five types of co-occurrences in Wikipedia to compute five association strength scores, which are then integrated by a linear weighted sum, where the weight parameters are trained using free association norms labeled by human beings.

These five types of co-occurrences are sentence level co-occurrences (*slc*), title link co-occurrences (*tlc*), title gloss co-occurrences (*tgc*), title body co-occurrences (*tbc*), and category level co-occurrences (*clc*). Examples of these co-occurrences are shown in Figure 1.

Specifically, *slc* refers to the co-occurrence of two terms in one sentence, such as *water* and *precipitation*. *tlc* refers to the co-occurrence of the page's title and anchor text in the page, such as *river* and *lake*, *river* and *precipitation*. *tgc* refers to the co-occurrence of a page's title and an unlinked term in the *gloss*, or the definition paragraph of this page, such as *river* and *stream*. *tbc* refers to the co-occurrence of a page's title and an unlinked term in the body paragraphs, the paragraphs except for gloss, such as *river* and *water*. *clc* refers to the number of categories in which two concepts co-occur, e.g., the concepts *Lake* and *Stream* share the category "Bodies of water". As described above, slc is between two terms, clc between two concepts, and the other three between a concept and a term. For all these types of co-occurrences, we first map the term or the concept to the corresponding super node, and then count the frequency.

[1] In this paper, we use "short text relatedness" and "short text similarity" interchangeably.

[2] The proof of this lemma is given at `http://adapt.seiee.sjtu.edu.cn/~keyang/assoc/`.

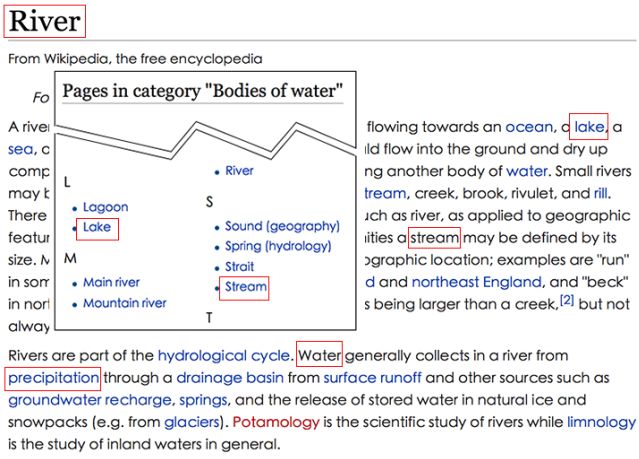

Figure 1: Five types of co-occurrences in Wikipedia

For each type of co-occurrences denoted as τ, where $\tau \in \{slc, tlc, tgc, tbc, clc\}$, we model the association strength from u to v after the measure proposed in (Wettler 1993) in (2). Here, α is an exponent parameter between 0 and 1. $p_\tau(u)$, $p_\tau(v)$ and $p_\tau(u,v)$ is computed as $\frac{f_\tau(\ )}{N_\tau}$, $\frac{f_\tau(\ )}{N_\tau}$ and $\frac{f_\tau(\ ,\ )}{N_\tau}$ respectively, where $f_\tau(u)$, $f_\tau(v)$ is the occurrence frequencies of u, v, $f_\tau(u,v)$ is the co-occurrence frequencies of u and v, and N_τ is the total number of tokens for a particular τ. We defer the discussion of the choice of α till Section 2.4.

$$r_\tau(u,v) = \frac{p_\tau(u,v)}{p_\tau(v)^\alpha p_\tau(u)} \tag{2}$$

$r_\tau(u,v)$ is normalized to $w_\tau(u,v)$:

$$w_\tau(u,v) = \frac{r_\tau(u,v)}{\sum\ r_\tau(u,v)} \tag{3}$$

We perform a case study to examine the different capabilities of capturing associated pairs by the five types of co-occurrences. We compare the normalized association strength $w_\tau(u,v)$ for every τ and the result is shown in Figure 2. u is set to be the super node of *river*, and v's are the super nodes of 5 terms most associated with *river*, as shown in Table 1. We observe the following: i) slc is distributed more uniformly among the pairs than others ii) only river-lake and river-stream have clc, as lake and stream are in the same type hierarchy as river iii) tlc, tgc and tbc capture the terms explaining or describing river, basically all the terms except for boat in this case. This shows that even though slc has been widely used in the literature, reflecting related terms of locality, other types of co-occurrences, though less studied, have complementary strength in terms of capturing associated pairs.

We then integrate the five types of $w_\tau(u,v)$ into a single strength score: $w(u,v) = \sum_\tau \theta_\tau w_\tau(u,v)$. We mimic human perception of relatedness in determining how to aggregate co-occurrences. Specifically, we train the weights θ_τ through a linear regression on the Florida Norms.

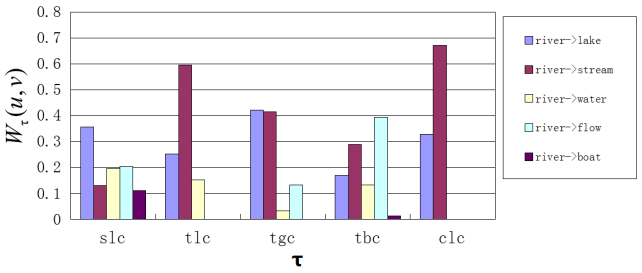

Figure 2: Comparison of $w_\tau(u,v)$ for different τ

We first use the terms appearing in Florida Norms as T_0 to determine the super node set V. Then we use every cue-response pair in Florida Norms as a training instance, with the label being the association strength computed form the norms. For every training instance we map the two terms into their corresponding super nodes u and v, and calculate $w_\tau(u,v)$ for every τ as its features. When performing linear regression, we impose the constraints that $\sum_\tau \theta_\tau = 1$, $0 < \theta_\tau < 1$ for all θ_τ, and that the intercept term must be equal to 0. The parameters θ_τ are then used to combine the different $w_\tau(u,v)$ regardless of the given T_0: As $\sum_\tau \theta_\tau = 1$ holds true, it is easy to attest that the combined strength score $w(u,v)$ meets the requirement of (1). An edge $e(u,v)$ exists if and only if $w(u,v) > 0$.

2.3 Relatedness computation

We now present how to leverage the constructed network for computing relatedness between terms and between short texts. In both algorithms, the intention is to leverage the latent *bridge* vertices between two observed vertices to provide supportive information in relatedness computation. The weight of a bridge vertex, with respect to an unordered pair of vertices $\{u,v\}$, is defined as in (4).

$$W_{\{\ ,\ \}}(x) = \max(w(u,x) \times w(x,v), w(v,x) \times w(x,u)) \tag{4}$$

For term relatedness computation, we first map any given term t to its corresponding vertex u in the association network. The relatedness between any vertex u and itself is always defined to be 1: $relate(u,u) = 1$. To compute relatedness between two vertices u and v, our baseline algorithm is to add up the weights of the edges between u and v:

$$relate(u,v) = w(u,v) + w(v,u) \tag{5}$$

This algorithm captures the intuition that if u associates more strongly to v or vice versa, u and v are often more related (see Table 1). As validated later in Section 3, despite its usefulness and intuitiveness in detecting related pairs, this algorithm leverages insufficient signals from the association network and hence obtains sub-optimal accuracy, especially when the association network is sparse. Thus, we propose a revised algorithm as a natural extension to the baseline:

$$relate(u,v) = w(u,v) + w(v,u) + \sum_{\in V} W_{\{\ ,\ \}}(x) \tag{6}$$

This algorithm captures the intuition that if u associates more strongly to v, directly or indirectly (via some bridge vertices), or vice versa, u and v are often more related.

For the short text relatedness computation, we abstract a given text as a bag of super nodes, i.e., a vector with each dimension being a super node and weight in that dimension being the occurrence frequency of the terms mapping to this super node. While cosine similarity could be directly computed to measure if two text vectors are similar or not, it suffers from low accuracy as semantically similar texts do not necessarily share identical or synonymous terms with each other. Therefore, we expand the original vectors before computing cosine similarity between two vectors, by adding bridge vertices identified through our association network as new dimensions. Algorithm 2 shows how we convert the original vector vec_0 to the expanded vector vec_+, where K is a parameter controlling the extent of the expansion (i.e., higher K means more expanded vertices).

Algorithm 2 Expand vector

```
1: function ExpandVector(network G(V, E), vector vec_0, integer K)
2:     vec_+ ← vec_0
3:     for u, v ∈dimension set of vec_0 and u ≠ v do
4:         V_K = top K vertices in V sorted by weights
5:         for x ∈ V_K do
6:             vec_+(x) ← vec_+(x) + 1
7:     return vec_+
```

2.4 Discussion

Instead of disambiguating the term occurring in a Wikipedia page to one of its concepts and defining each vertex to be a disambiguated concept in the association network, we choose to define each vertex to be a super node, comprising multiple concepts for a term. That is because, even though it is possible to disambiguate a term in Wikipedia pages by taking advantage of contextual information, such a task is more difficult on the free association norms, where virtually no context is available. Even worse, the two end-to-end tasks (term and short text relatedness) also inherently lack context information to perform reliable disambiguation.

When computing the association strength between two super nodes u and v, the parameter α needs to be chosen to instantiate the general form shown in (2). One natural choice is to set α to be 0, which turns the formula into conditional probability, i.e., the probability of observing v, given u. However, it is argued previously (Wettler 1993; Washtell 2009) that the conditional probability measure does not take into consideration the general frequency of the response word and therefore tends to bias toward highly frequent words, such as function words. As a result, we follow (Wettler 1993) to set α to be 0.66, which, according to them, perform the best in estimating word association.

Our algorithms for relatedness computation are for showcasing the power of the association network, and thus many other algorithms can be developed to take advantage of the full potential of the association network.

3 Experimental Results

This section primarily evaluates two association networks, one constructed only using the original free association norms (denoted as $AN_{f\ ee}$), and the other constructed through the approach proposed in Section 2 (denoted as $AN_{\ i\ i}$). The results on $AN_{f\ ee}$ show the usefulness as well as limitations of free association norms, while the results on $AN_{\ i\ i}$ validate the added benefits of Wikipedia structures in working around the two limitations of $AN_{f\ ee}$, leading to better performance in semantic relatedness computation tasks. [3]

3.1 Data sources and statistics

The original Florida free association norms data contains 5,019 cue words (which form the set of *normed words*) and a total of 72,176 cue-response pairs. 63,619 of these pairs contain responses that are also normed words. These pairs are called *normed pairs* with known forward (cue-to-target) and backward (from target-to-cue) strengths.

Our baseline association network, $AN_{f\ ee}$, is made up of the 5019 normed words as vertices and the 63,619 normed pairs as directed edges. Each edge carries a normalized weight $w(u, v)$, which is proportional to $Pr(v \mid u)$, Note, in $AN_{f\ ee}$, each word forms a super node by itself, as we aim to evaluate usefulness of the original free association norms, without depending on additional knowledge (e.g., Wikipedia) to construct super nodes.

Our proposed synthetic association network, $AN_{\ i\ i}$, consists of 17,469 vertices (super nodes) and 107M directed edges. This network is constructed using the 20,000 most common English words (with stop words removed) as given T_0, and using a Wikipedia dump from July, 2014.

Our test set for evaluting term relatedness is the well-known *WordSimilarity-353* (Finkelstein et al. 2002) (a.k.a. WS-353 with 353 word pairs),

For testing short text similarity, we use the well-known public set *Li30* (Li et al. 2006), comprising 30 pairs of short texts. A newly constructed dataset STSS-131 (O'shea, Bandar, and Crockett 2013) is used to tune the parameter K decribed in Algorithm 2.

3.2 AN_{free} v.s. AN_{wiki}

To illustrate the usefulness of free association network, as well as its limitation in semantic relatedness computation, we create WS-227, a subset of WS-353, in which all words belong to some vertex in $AN_{f\ ee}$.

The baseline algorithm with (5) as its metric is denoted by AN^0, while the revised algorithm with (6) as its metric is denoted by AN^+. We apply AN^0 and AN^+ using $AN_{f\ ee}$ and $AN_{\ i\ i}$ on WS-227 and WS-353, and compare the performance measured in Spearman correlation with two other well-known algorithms, namely LSA (Deerwester et al. 1990) and ESA (Gabrilovich and Markovitch 2007), in Table 2. The result for LSA is obtained from the widely

[3] A demo of our system is available at `http://adapt.seiee.sjtu.edu.cn/~keyang/assoc/`.

used online portal[4], while the result for ESA is obtained from ESAlib[5].

We observe the following: 1) $AN^{+}_{f\ ee}$ performs better than LSA and ESA on WS-227, despite its relatively small size, which suggests that free association can be useful in computing semantic relatedness. 2) However, when tested on WS-353, due to its limited vocabulary, $AN^{+}_{f\ ee}$ shows a drastic degrade in performance, which reflects one of its primary limitations. Conversely, $AN_{\ i\ i}$, of a larger lexical coverage, performs consistently well on both WS-227 and WS-353. 3) Due to $AN_{f\ ee}$'s another limitation, sparseness, $AN^{0}_{f\ ee}$ exhibits sub-optimal performance on both WS-227 and WS-353; while $AN^{+}_{f\ ee}$ shows a significant improvement as the sparseness problem is alleviated by leveraging the latent bridge vertices. 4) Though the best performance is obtained by $AN^{+}_{i\ i}$, its advantage over $AN^{0}_{i\ i}$ is not large. We argue that it is because by reverse-engineering the association strength into an aggregation function of a vector of structured co-occurrence, $AN_{\ i\ i}$ alleviates sparseness by enabling to infer the edge weights missing in $AN_{f\ ee}$.

Table 2: Spearman correlation on two WS datasets

Methods	WS-227	WS-353
LSA	0.542	0.579
ESA	0.727	0.744
$AN^{0}_{f\ ee}$	0.645	0.476
$AN^{+}_{f\ ee}$	0.752	0.512
$AN^{0}_{i\ i}$	0.758	0.785
$AN^{+}_{i\ i}$	**0.782**	**0.813**

3.3 Prediction of free association

In this experiment, we evaluate if $AN_{\ i\ i}$ can be used to predict free association strengths given by humans. We compute Spearman correlation between scores predicted by a number of competing methods (Washtell 2009) and the human association strength computed from the Kent's free association norms (1910). Our method is just mapping the two terms to vertex u and v, and assigning $w(u, v)$ as predicted association strength.

As is shown in Table 3, $AN_{\ i\ i}$ does a reasonable job in simulating free association, compared with other common approaches. And as a reference, the Spearman correlation between the human labeled scores of Kent dataset and those of the Minnesota dataset (Jenkins 1970) using the same set of cue words is 0.4, which can be viewed as an upper bound for computer-based systems.

3.4 End-to-end tasks: term & short text relatedness

Table 4 compares $AN_{\ i\ i}$ with a number of previous approaches on the term relatedness computation using WS-353 dataset. Our association network achieves state-of-the-art results on correlation with human scores.

[4]http://lsa.colorado.edu/

[5]http://ticcky.github.io/esalib/

Table 3: Association Prediction

Methods	Spearman
Cond. Prob.	0.31
SCI	0.34
PMI	0.28
Dice/Jaccard	0.32
$AN_{\ i\ i}$	**0.37**

Table 4: Spearman correlation on WS-353 dataset

Methods	Spearman
Resnik (1995)	0.353
LSA-Landauer (1997)	0.581
Lin (1998)	0.348
Roget-Jarmasz (2003)	0.415
ESA-Gabrilovich (2007)	0.75
Agirre (2009)	0.78
Reisinger (2010)	0.77
SSA-Hassan (2011)	0.629
TSA-Radinsky (2011)	0.80
CLEAR-Halawi (2012)	0.810
Xu (2014)	0.683
$AN_{\ i\ i}$	**0.813**

Table 5 shows that our association network outperforms all existing approaches by significant margins on short text similarity task.

Recall that, Algorithm 2 is parameterized by K determining the extent of expansion. Our reported results use $K = 10$, empirically tuned based on STSS-131 dataset.

Table 5: Pearson and Spearman correlation on Li30 dataset

Methods	Pearson	Spearman
STASIS-Li (2006)	0.816	0.813
LIU (2007)	0.841	0.854
LSA-OShea (2008)	0.838	0.871
STS-Islam (2008)	0.853	0.838
Omiotis-Tsatsaronis (2010)	0.856	0.891
WSD-STS-Ho (2010)	0.864	0.834
SPD-STS-Ho (2010)	0.895	0.903
SSA-Hassan (2011)	0.881	0.878
LDA-Guo (2012)	0.842	0.866
WTMF-Guo (2012)	0.898	0.909
WTMF+PK-Guo (2013)	0.902	–
$AN_{\ i\ i}$	**0.942**	**0.940**

3.5 Effects of different co-occurrences and free association training

In this experiment, we compare an association network built from only sentence-level co-occurrences (slc), an association network with 5 types of co-occurrences uniformly combined (uniform), and our proposed network, which comes with weights trained from free association norms. Table 6 gives rise to these observations: i) $AN_{\ i\ i}$(uniform) outperforms $AN_{\ i\ i}$(slc) by a large margin on both tasks, which

shows that the four additional types of co-occurrences are useful in capturing signals not available in slc; ii) $AN_{\ i\ i}$ further improves the results from $AN_{\ i\ i}$(uniform) by a substantial margin, which shows that signals tapped from free association norms can indeed benefit semantic relatedness computation tasks.

Table 6: Several variants of $AN_{\ i\ i}$

Methods	WS-353	Li30
$AN_{\ i\ i}$(slc)	0.734	0.884
$AN_{\ i\ i}$(uniform)	0.766	0.903
$AN_{\ i\ i}$	**0.813**	**0.942**

3.6 Execution time

Average execution time for computing the relatedness score for a pair of terms in WS-353 is $10.3ms$, and for a pair of short texts in Li30 is $465.3ms$. The time and space consumption can be further reduced by filtering out edges with insignificant weights. Experiments show that by removing up to 90% of the edges, the accuracy in both term and short text relatedness remains virtually constant, and at the same time the execution time for a pair of terms and a pair of short texts are reduced to $1.4ms$ and $19.1ms$, respectively.

4 Related Work

In this section, we introduce a number of studies in *semantic relatedness computation* and related work in *free association*.

Previous approaches to semantic relatedness pursue two main directions, of using hand-crafted lexical taxonomies like WordNet (Miller 1995) or Roget's Thesaurus (Roget 1911) as semantic knowledge, or of employing probabilistic approaches to decode semantics based on large corpora.

The first approach of using hand-crafted resources proposes knowledge-based measures that tap into the properties of their underlying structure to compute semantic relatedness (Roget 1911; Lin 1998; Leacock and Chodorow 1998; Hirst and St-Onge 1998; Jiang and Conrath 1997; Resnik 1995; Wu and Palmer 1994). Though showing potential in such tasks like term relatedness computation, this approach requires to construct manually curated lexical resources and thus cannot easily scale to larger lexical coverage or to a new language.

On the other hand, the second approach of using corpus-based measures, instead of relying on human-organized knowledge, utilize the contextual information and patterns observed in large corpus to construct semantic profiles for words. Latent Semantic Analysis (LSA) (Deerwester et al. 1990) was an original approach to leverage word co-occurrences from a large corpus of text, and "learns" its representation by applying Singular Value Decomposition to the words-by-documents co-occurrence matrix. Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2007) as well as Salient Semantic Analysis (SSA) (Hassan and Mihalcea 2011) were proposed to incorporate large amounts of human knowledge such as Wikipedia into word relatedness computation. They both represent a word as a concept vector, where each dimension corresponds to a Wikipedia concept. Later, Temporal Semantic Analysis (TSA) (Radinsky et al. 2011) considered that words have different meanings over time and extended the concept vector with a temporal dimension. To bridge the corpus-based measures with knowledge-based measures, Constrained LEArning of Relatedness (CLEAR) (Halawi et al. 2012) was proposed to learn word relatedness based on word occurrence statistics from large corpora while constraining the learning process by incorporating knowledge from WordNet. Some recent works like (Mikolov et al. 2013) used machine learning techniques to compute continuous vector representations of words from large datasets , shown to perform better than LSA for preserving linear regularities among words.

Some models aim particularly at solving the similarity problem between two sentences, or two short texts (Guo and Diab 2012; 2013; Ho et al. 2010; OShea et al. 2008). WSD-Based Sentence Similarity (Ho et al. 2010) was proposed to compute the similarity between two sentences based on a comparison of their actual meanings by integrating word sense disambiguation. WTMF (Guo and Diab 2012) was proposed to model the missing words in the sentences as a typically overlooked feature to address the sparseness problem for the short text similarity task.

All the existing semantic relatedness models mentioned above, though leveraging some useful signals from hand-crafted lexical taxonomies or large corpus text, fail to actively take advantage of the human perception signal in semantic relatedness computation. Our approach, by effectively bridging this gap using signals in the well-studied psychological process of free association, outperforms state-of-the-art models in both word and short text relatedness tasks.

Free association is a task requiring human participants to produce the easily associated word for the given cue word, to tap into human perception acquired through world experience (Nelson, McEvoy, and Schreiber 2004). Mining the signals contained in this cognitive process is made possible by several collections of free association norms (Nelson, McEvoy, and Schreiber 2004; Kent and Rosanoff 1910; Jenkins 1970; Kiss et al. 1973), which are typically collected by researchers in phychology and cognitive science. As the Florida Norms is the largest collection available, and also the most recent in time, we choose to use it as our primary source of human perception to be combined with signals from Wikipedia.

5 Conclusion

We synthetically build an association network, by aggregating Wikipedia signals, using free association as a training data. Our evaluation results validated that our proposed framework reaches state-of-the-art in a standard benchmark for term relatedness computation and outperforms all other state-of-the-arts for short text similarity computation by a significant margin.

Acknowledgement

Kenny Q. Zhu, the contact author, was supported by NSFC grant 61373031 and NSFC-NRF Joint Research Program. Seung-won Hwang was supported under the framework of international cooperation program managed by NRF of Korea (2014K2A2A2000519). This work received contributions from Kailang Jiang and Jinyi Lu.

References

Agirre, E.; Soroa, A.; Alfonseca, E.; Hall, K.; Kravalova, J.; and Pasca, M. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL'09*.

Budanitsky, A., and Hirst, G. 2006. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics* 32.

Deerwester, S. C.; Dumais, S. T.; Landauer, T. K.; Furnas, G. W.; and Harshman, R. A. 1990. Indexing by latent semantic analysis. *JASIS* 41(6).

Finkelstein, L.; Gabrilovich, E.; Matias, Y.; Rivlin, E.; Solan, Z.; Wolfman, G.; and Ruppin, E. 2002. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.* 20(1).

Gabrilovich, E., and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*.

Guo, W., and Diab, M. 2012. Modeling sentences in the latent space. In *ACL*.

Guo, W., and Diab, M. T. 2013. Improving lexical semantics for sentential semantics: Modeling selectional preference and similar words in a latent variable model. In *HLT-NAACL*.

Halawi, G.; Dror, G.; Gabrilovich, E.; and Koren, Y. 2012. Large-scale learning of word relatedness with constraints. In *KDD*.

Hassan, S., and Mihalcea, R. 2011. Semantic relatedness using salient semantic analysis. In *AAAI*.

Hirst, G., and St-Onge, D. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. In *WordNet: An Electronic Lexical Database*.

Ho, C.; Murad, M. A. A.; Kadir, R. A.; and Doraisamy, S. C. 2010. Word sense disambiguation-based sentence similarity. In *ACL*.

Islam, A., and Inkpen, D. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *TKDD* 2(2).

Jarmasz, M. 2003. *Roget's Thesaurus as a Lexical Resource for Natural Language Processing*. Ph.D. Dissertation, Ottawa-Carleton Institute for Computer Science.

Jenkins, J. J. 1970. The 1952 minnesota word association norms. *Norms of word association*.

Jiang, J., and Conrath, D. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Intl. Conf. on Res. in Computational Linguistics*.

Kent, G. H., and Rosanoff, A. J. 1910. *A study of association in insanity*. American Journal of Insanity.

Kiss, G. R.; Armstrong, C.; Milroy, R.; and Piper, J. 1973. An associative thesaurus of english and its computer analysis. *The computer and literary studies*.

Landauer, T. K.; Laham, D.; Rehder, B.; and Schreiner, M. E. 1997. How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. In *CogSci*.

Leacock, C., and Chodorow, M. 1998. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49(2).

Li, Y.; McLean, D.; Bandar, Z. A.; O'shea, J. D.; and Crockett, K. 2006. Sentence similarity based on semantic nets and corpus statistics. *TKDE* 18(8).

Lin, D. 1998. An information-theoretic definition of similarity. In *ICML'98*.

Liu, X.; Zhou, Y.; and Zheng, R. 2007. Sentence similarity based on dynamic time warping. In *ICSC*.

McGill, M. 1979. An evaluation of factors affecting document ranking by information retrieval systems. Tech. Rep., Syracuse Univ , School of Information Studies.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, G. A. 1995. WordNet: A lexical database for english. *Commun. ACM* 38(11).

Nelson, D. L.; McEvoy, C. L.; and Schreiber, T. A. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Res. Methods, Instruments, & Computers* 36(3).

O'shea, J.; Bandar, Z.; and Crockett, K. 2013. A new benchmark dataset with production methodology for short text semantic similarity algorithms. *TSLP* 10(4).

OShea, J.; Bandar, Z.; Crockett, K.; and McLean, D. 2008. A comparative study of two short text semantic similarity measures. In *KES-AMSTA*.

Radinsky, K.; Agichtein, E.; Gabrilovich, E.; and Markovitch, S. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *WWW*.

Reisinger, J., and Mooney, R. J. 2010. Multi-prototype vector-space models of word meaning. In *NAACL*.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI'95*.

Roget, P. M. 1911. *Roget's Thesaurus of English Words and Phrases*. HarperCollins New York.

Tsatsaronis, G.; Varlamis, I.; and Vazirgiannis, M. 2010. Text relatedness based on a word thesaurus. *JAIR* 37(1).

Washtell, J. 2009. Co-dispersion: A windowless approach to lexical association. In *EACL*.

Wettler, M, R. R. 1993. Computation of word associations based on the co-occurrences of words in large corpora. In *the 1st Workshop on Very Large Corpora*.

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *ACL*.

Xu, C.; Bai, Y.; Bian, J.; Gao, B.; Wang, G.; Liu, X.; and Liu, T.-Y. 2014. Rc-net: A general framework for incorporating knowledge into word representations. In *CIKM*.