A Word at a Time: Computing Word Relatedness using Temporal Semantic Analysis

Kira Radinsky*, Eugene Agichtein†, Evgeniy Gabrilovich‡, Shaul Markovitch*

* CS Department, Technion—Israel Institute of Technology, 32000 Haifa, Israel
† Math & CS Department, Emory University, 400 Dowman Drive, Atlanta, GA 30322, USA
‡ Yahoo! Research, 4301 Great America Parkway, Santa Clara, CA 95054, USA

* {kirar|shaulm}@cs.technion.ac.il † eugene@mathcs.emory.edu ‡ gabr@yahoo-inc.com

ABSTRACT

Computing the degree of semantic relatedness of words is a key functionality of many language applications such as search, clustering, and disambiguation. Previous approaches to computing semantic relatedness mostly used static language resources, while essentially ignoring their temporal aspects. We believe that a considerable amount of relatedness information can also be found in studying patterns of word usage over time. Consider, for instance, a newspaper archive spanning many years. Two words such as "war" and "peace" might rarely co-occur in the same articles, yet their patterns of use over time might be similar. In this paper, we propose a new semantic relatedness model, Temporal Semantic Analysis (TSA), which captures this temporal information. The previous state of the art method, Explicit Semantic Analysis (ESA), represented word semantics as a vector of concepts. TSA uses a more refined representation, where each concept is no longer scalar, but is instead represented as time series over a corpus of temporally-ordered documents. To the best of our knowledge, this is the first attempt to incorporate temporal evidence into models of semantic relatedness. Empirical evaluation shows that TSA provides consistent improvements over the state of the art ESA results on multiple benchmarks.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.5.4 [**Pattern Recognition**]: Applications

General Terms

Algorithms, Experimentation

Keywords

temporal dynamics, temporal semantics, semantic analysis, word relatedness, semantic similarity

1. INTRODUCTION

The ability to quantify semantic relatedness of texts underlies many fundamental tasks in natural language processing, including information retrieval, word sense disambiguation, text clustering, and error correction. Previous approaches to computing semantic relatedness used various linguistic resources, such as WordNet, Wikipedia, or large-scale text corpora for methods like Latent Semantic Analysis (LSA). Yet all of these approaches essentially considered the underlying linguistic resource as a static collection of texts or concepts. In this paper we argue that there is an additional source of rich information about semantic relatedness of words, which can be revealed by studying the patterns of word occurrence over time.

Consider, for example, words such as "war" and "peace". While these words are clearly related, they might rarely be mentioned in the same documents. However, they are likely to be mentioned roughly around the same time (say, in different articles posted during the same day, or in adjacent days). In this work, we use the New York Times archive spanning over 130 years. For each word, we construct the time series of its occurrence in New York Times articles. We posit that if there is a correlation between the time series of two words, then the meanings of the two words are related.

In principle, there are a number of cases when temporal information could offer a complementary source of signal, which is not captured by other models. Synonyms (that is, words with similar meanings) are rarely used in the same article since an author usually sticks to one set of terms, yet they can be used by different authors in different articles describing the same events. Looking at their coordination in time allows us to leverage the opinions of multiple authors collectively. As another example, consider pairs of words that form stock phrases, such as "luxury car". Taken individually, the two words in each pair have very different meanings, and are likely to be judged as such by existing methods. On the other hand, these words are indeed related, and the frequency of their use over time exhibits non-trivial correlation. Especially interesting are pairs of words that have implicit relationships such as "war" and "peace" or "stock" and "oil", which tend to correlate in frequency of use over time. Figures 1 and 2 depict these correlations in time. The proposed method, Temporal Semantic Analysis, captures such correlations, and is able to better estimate semantic relatedness than methods that only use static snapshots of linguistic resources.

The contributions of this paper are threefold. First, we propose to use temporal information as a complementary source of signal to detect semantic relatedness of words. Specifically, we introduce *Temporal Semantic Analysis* (TSA), which leverages this information and computes a refined metric of semantic relatedness. Second, we construct a new dataset for semantic relatedness of words, which we have judged with the help of Amazon's Mechanical Turk service.

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

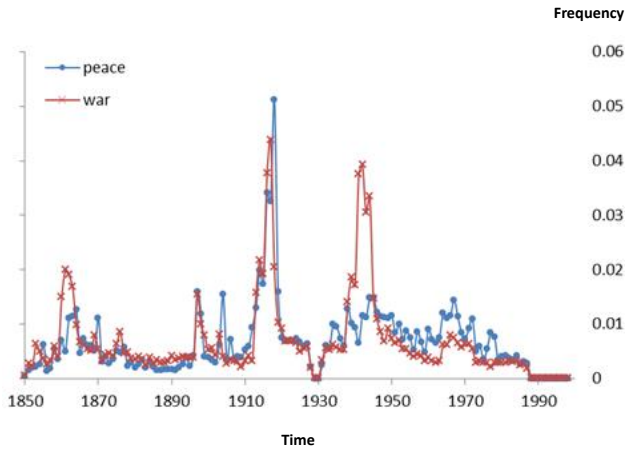

Figure 1: Time series (1870-1988) of the words "war" (red) and "peace"(blue). The words correlate over time.

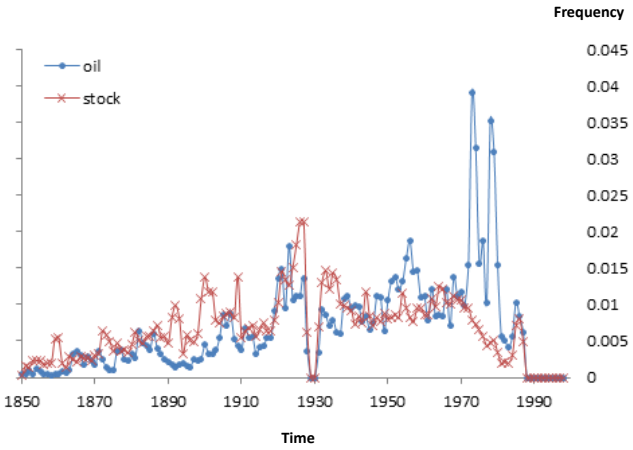

Figure 2: Time series (1870-1988) of the words "stock" (red) and "oil"(blue). The words correlate over time.

In contrast with the previous standard benchmark, WS-353, our new dataset has been constructed by a computer algorithm (also presented below), which eliminates subjective selection of words. We make the new dataset publicly available for further research in the field. Finally, empirical evaluation shows that TSA exhibits superior performance compared to the previous state of the art method (ESA), and achieves higher correlation with human judgments on both datasets.

2. TEMPORAL SEMANTIC ANALYSIS

We propose Temporal Semantic Analysis (TSA), which is composed of two novel components: a new approach, described in this section, for *representing* the semantics of natural language words, and a new method, described in Section 3, for *computing* the semantic relatedness between words.

Our method is based on associating each word with a weighted *vector of concepts*. Such concepts can be derived from crowd intelligence folksonomies such as Wikipedia, cotagging in Flickr, or from online bookmarking services such as del.icio.us. This is similar to recent semantics approaches such as ESA [16]. However, while ESA uses a *static* representation of each concept, we use the concept *dynamics*—its behavior over time, represented by the time series of the concept occurrence. Thus, instead of representing a word with a vector of unit concepts, vectors of time series are manipulated, where each time series describes concept dynamics over time. Our hypothesis is that *concepts that behave similarly over time, are semantically related*. Such a rich representation of words (adding the extra temporal dimension) could facilitate the discovery of implicit semantic relationships between the original words. As we will show experimentally, the naive approach of directly computing temporal correlation between words (without the concept vector representation) is not effective.

Thus, our TSA method consists of three main steps:

1. **Represent words as concept vectors**: using a concept repository of choice (e.g., Wikipedia or Flickr image tags), represent a word as a set of associated concepts with weights (Section 2.1).

2. **Extract temporal dynamics for each concept**: using a corpus of choice (e.g., New York Times archive), quantify concept occurrence for each time period (e.g., a day) and build its time series (Section 2.2).

3. **Extending static representation with temporal dynamics**: finally, scale each concept's time series according to the concept's original weight from item 1 above (Section 2.3).

2.1 Representing Words as Concept Vectors

In our representation, each word is mapped into a vector of concepts — a *concept vector*. For each concept a static weight is computed. We consider several such representations over multiple folksonomies:

1. Wikipedia Concepts — Wikipedia is among the largest knowledge repositories on the Web, which is written collaboratively by millions of volunteers around the world, and almost all of its articles can be edited by any user. Wikipedia is available in dozens of languages, while its English version is the largest of all with more than 500 million words in over three million articles. Vector space models based on this ontology have been used by many works for semantic relatedness [16, 32]. In those representations, each entry in the concept vector is a TFIDF-based function of the strength of association between the word and the concept in Wikipedia.

2. Flickr Image Tags — Flickr is an online image hosting community. Photo submitters have the option to add metadata to each image in the forms of natural language tags. This feature enables searchers to find images related to specific topics. The natural concepts in this representation are the Flickr tags.

3. Del.icio.us Bookmarks — del.icio.us is a social URL bookmarking service, with the possibility to search and explore new bookmarks. The service had, by the end of 2008, more than 5.3 million users and 180 million unique bookmarked URLs. Del.icio.us users can tag each of their bookmarks with free text, and we use these tags as concepts.

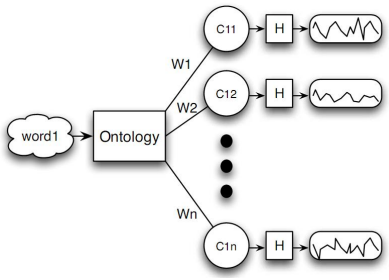

Figure 3: Each word is represented by a weighted vector of concept time series (produced from a historical archive H). The weight w_i of each concept corresponds to the concept "importance" w.r.t. the original word.

2.2 Temporal Concept Dynamics

Let c be a concept represented by a sequence of words $wc_1, \ldots, wc_k$. Let d be a document. We say that c *appears* in d if its words appear in the document with a distance of at most ε words between each pair wc_i, wc_j, where ε is a proximity relaxation parameter (in the experiments we set $\varepsilon = 20$). That is, a concept *appears* in a document if there is a window of size ε where all the concept words appear. For example, for the concept c — "Great Fire of London" — we say that the c appears in a document d, if the words "Great", "Fire", "of", "London" appear in the document with a distance of at most ε between each word.

Let $t_1, \ldots, t_n$ be a sequence of consecutive discrete time points (e.g., days). Let $H = D_1, \ldots, D_n$ be a *history* represented by a set of document collections, where D_i is a collection of documents associated with time t_i. We define the *dynamics* of a concept c to be the time series of its frequency of appearance in H:

$$Dynamics(c) = \langle \frac{|\{d \in D_1 | appears(c,d)\}|}{|D_1|}, \ldots, \frac{|\{d \in D_n | appears(c,d)\}|}{|D_n|} \rangle \quad (1)$$

In the experiments described in this paper we used New York Times articles since 1870 for history. Each time point is a day, and the collection of documents associated with a day is the set of articles appearing on that day.

2.3 Extending Static Representation with Temporal Signals

Our approach is inspired by the desire to augment text representation with massive amounts of temporal world knowledge. Hence, we represent a word as a weighted mixture of concept time series, where the weights correspond to the concept "importance" w.r.t. the original word (Figure 3).

In common semantic representations (such as ESA [16]) a word is represented as a weighted vector of concepts (derived from Wikipedia articles). In ESA, each vector entry contains a single (static) TFIDF weight, which expresses the strength of association of the word and the concept. Our TSA method extends ESA so that each entry in the vector corresponds to a time series, computed as described above.

Procedure SEMANTIC RELATEDNESS(t_1, t_2)
(1)$C(t_1) = \{ts_1^1, \ldots, ts_n^1\}$
(2)$C(t_2) = \{ts_1^2, \ldots, ts_m^2\}$
(3)$R(t_1, t_2) \leftarrow 0$
(4)**Repeat** $Min(m, n)$ **times**
(5) $\langle \hat{ts_1}, \hat{ts_2} \rangle = arg\max_{\langle ts_1, ts_2 \rangle \in C(t_1) \times C(t_2)} Q(ts_1, ts_2)$
(6) $R(t_1, t_2) \leftarrow R(t_1, t_2) + Q(\hat{ts_1}, \hat{ts_2})$
(7) $C(t_1) \leftarrow C(t_1) \backslash \{\hat{ts_1}\}$
(8) $C(t_2) \leftarrow C(t_2) \backslash \{\hat{ts_2}\}$
(9)**Return** $R(t_1, t_2)$

Figure 4: A greedy algorithm for computing the semantic relatedness between two words. The procedure assumes the availability of a function Q that determines relatedness between a pair of time series ts_i associated with two concepts.

3. USING TSA FOR COMPUTING SEMANTIC RELATEDNESS

To compute semantic relatedness of a pair of words we compare their vectors (as defined in Section 2.3) using measurements of weighted distance between multiple time series, combined with the static semantic similarity measure of the concepts. This approach, therefore, integrates both temporal and static semantic behavior of the words.

3.1 TSA-based Semantic-Relatedness Algorithm

The ESA method for computing semantic relatedness is based on the assumption that related words share highly-weighted concepts in their representations. The TSA approach does not assume so. We only assume that highly-weighted concepts of the related words are *related*.

Suppose we are trying to find the relatedness between words t_1 and t_2. Assume that t_1 is mapped to a set of concepts $C(t_1) = \{c_1^1, \ldots, c_n^1\}$ and t_2 is mapped to $C(t_2) = \{c_1^2, \ldots, c_m^2\}$. Suppose we have a function Q that determines relatedness between two individual concepts using their dynamics (as defined in Section 2.2). Assuming w.l.o.g $n \leq m$, we can define the relatedness R between t_1 and t_2 as the maximal sum of pairwise concept relatedness over all ordered subsets of size n of $C(t_2)$:

$$R(t_1, t_2) = \max_{j_l \in (1 \ldots \binom{m}{n})} \sum_{l=1,\ldots,n} Q(c_l^1, c_{j_l}^2) \quad (2)$$

This exhaustive search over all possible pairs is, however, infeasible. Therefore we take an alternative greedy approach, which is formally described in Figure 4. The procedure at each step finds a pair of time series with the highest relatedness Q (line 5 in the algorithm), removes them and proceeds (lines 7 and 8). Iteratively, the relatedness $R(t_1, t_2)$ is computed as the sum of relatedness of the matching concepts (line 6). This procedure complexity is $O(n \cdot m \cdot max(|ts|))$, where $|ts|$ is the length of the time series representing the concepts.

3.2 Similarity Between Individual Time Series

The relatedness Q between two concepts is determined by comparing their dynamics. Our basic assumption is that related concepts correlate in their temporal behavior. For comparing the concepts associated time series, we use two

existing methods for measuring time series similarity — cross correlation and dynamic time wrapping (DTW).

3.2.1 Cross Correlation

In statistics, cross correlation is a method for measuring statistical relations, e.g., measuring similarity of two random variables. A common measurement for this purpose is the Pearson's product-moment coefficient which is defined as:

$$\text{corr}(X,Y) = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-E(X))(Y-E(Y))]}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\overline{y})^2}} \quad (3)$$

In signal processing, cross-correlation is used as a measure of similarity of two signals as a function of a time-lag applied on one of the signals — a variation of the Pearson coefficient to different time delays between two time series in Figure 5. An innate characteristic of this measure is identification of similar time series in volume, with consideration of time shifts. In our representation, where words are represented as time series, words whose frequencies correlate in volume, but with a time lag, will be identified as similar. When we wish to evaluate the correlation of the two words' time-series, we compare the time series starting from the first time point they both started appearing, until the time point when one of the words stopped appearing. For example, the word "computer" did not appear during the 1800s, and started to appear only around 1930. Therefore, when we compare it to the word "radio", we calculate the cross correlation only during the period starting at 1930.

3.2.2 Dynamic Time Warping

The DTW algorithm [5] measures the similarity between two time series that may differ in time scale, but similar in shape. In speech recognition, this method is used to identify similar sounds between different speakers whose speech speed and pitch might be different. The algorithm defines a local cost matrix $C \in R^{|ts_1|\times|ts_2|}$ of two time series ts_1 and ts_2 as

$$C_{i,j} = \|ts_1[i] - ts_2[j]\|, i \in \langle 1 \ldots |ts_1|\rangle, j \in \langle 1 \ldots |ts_2|\rangle \quad (4)$$

where $\|ts_1[i]-ts_2[j]\|$ is a distance metric between two points of the time series.

Given this cost matrix, DTW constructs an alignment path that minimizes the cost over this cost matrix. This alignment p is called the "warping path", and defined as a sequence of points pairs $p = (pair_1, \ldots pair_k)$, where $pair_l = (i,j) \in \langle 1 \ldots |ts_1|\rangle \times \langle 1 \ldots |ts_2|\rangle$ is a pair of indexes in ts_1 and ts_2 respectively. Each consequent pair preserves the ordering of the points in ts_1 and ts_2, and enforces the first and last points of the warping path to be the first and last points of ts_1 and ts_2. For each warping path p we compute its cost as $c(p) = \sum_{l=1}^{k} C(pair_l)$. The DTW is defined to be the minimum optimal warping path

$$DTW(ts_1, ts_2) = min\{c(p)|p \in P^{|ts_1|\times|ts_2|}\} \quad (5)$$

where P are all possible warping paths. A dynamic programming algorithm (similar to the one in Figure 6) is usually applied to compute the optimal warping path of the two sequences.

This similarity measurement, as opposed to time series cross-correlation distance (cf. Section 3.2.1) is much more flexible, hence we decided to experiment with it as well.

3.2.3 Temporal Weighting Function

As the meaning of the words changes over time, more recent concept correlation are more significant than past correlation. Therefore, when measuring the distance between two individual time series, higher weights to recent similarities should be given. We apply several linear and non-linear weighting functions to the above time series distance functions (see Section 5.2.4). Let $f(i,j)$ be such a function, whose parameters are two time points i, j. Thus, we modify DTW definition of $\|ts_1(i) - ts_2(j)\|$ to

$$\|ts_1(i) - ts_2(j)\| \cdot f(i,j) \quad (6)$$

and the covariance definition in the cross-correlation distance now changes to

$$cov(ts_1, ts_2) \leftarrow cov(ts_1, ts_2) + f(i,j) \cdot [(ts_1[index] - E(ts_1)) \cdot (ts_2[delayedIndex] - -E(ts_2))] \quad (7)$$

We described how our TSA method represents words as concepts (Section 3.1), and how the temporal dynamics of the concept usage over time can be used to compute semantic relatedness (Section 3.2). We now turn to the experimental evaluation of our approach.

4. EXPERIMENTAL SETUP

We implemented our TSA approach using the New York Times archive (1863-2004). For each day we had an average of 50 abstracts of articles, which after parsing yielded 1.42 GB of texts with a total of 565,540 distinct words. In this section we describe the methodology we used in our experiments and then describe a novel algorithm for automatically creating benchmarks for word relatedness tasks.

Both ESA and TSA were implemented on the concepts extracted from the folksonomies presented in Section 2.1, and therefore use the same vector representations. This will allow us to isolate the performance of the temporal dimension in the TSA semantics.

4.1 Experimental Methodology

Methods compared: We compare our algorithm and representations to the state of the art semantic representation — Explicit Semantic Analysis (ESA), which has been shown to be significantly superior to other approaches [16]. This approach projects words into a high-dimensional space of concepts derived from Wikipedia. Using machine learning techniques, it represent the meaning of a word as a weighted vector of Wikipedia-based concepts. Each concept in the vector is weighted by relevance to the word. Assessing the relatedness of words in this space is done by utilizing cosine distance – a conventional metric of comparison of high-dimensional vectors.

Evaluation metrics: As in prior published studies, in our evaluation we use Spearman correlation coefficient to compare the predicted relatedness scores with human judgements. The comparison is applied on both our algorithm and representations and the current state of the art.

Statistical Significance: We compare the rank correlation coefficient of our method, $rank_1$, to the competitive methods

Procedure CROSS CORRELATION(ts_1, ts_2)
(1)$similarity(ts_1, ts_2) = 0$
(2)$cov(ts_1, ts_2) = 0$
(3)**For** $delay = \{-delay_{min} \dots delay_{max}\}$
(4) **For** $index = \{0 \dots Min(|ts_1|, |ts_2|)\}$
(5) $delayedIndex = index + delay$
(6) $cov(ts_1, ts_2) \leftarrow cov(ts_1, ts_2) + (ts_1[index] - E(ts_1)) * (ts_2[delayedIndex] - E(ts_2))$
(7) $corr@delay(ts_1, ts_2) \leftarrow \frac{cov(ts_1, ts_2)}{N \cdot \sigma_{ts_1} \sigma_{ts_2}}$
(8) $similarity(ts_1, ts_2) \leftarrow Max(similarity(ts_1, ts_2), corr@delay(ts_1, ts_2))$
(9)**Return** $similarity(ts_1, ts_2)$

Figure 5: Time series cross correlation

Procedure DTW(ts_1, ts_2, C)
(1)$n \leftarrow Min(|ts_1|, |ts_2|)$
(2)$dtw(ts_1, ts_2) \leftarrow$ new $[|ts_1| \times |ts_2|]$
(3)**For** $i = \{1 \dots n\}$
(4) $dtw(i, 1) \leftarrow dtw(i-1, 1) + c(i, 1)$
(5) $dtw(1, i) \leftarrow dtw(1, i-1) + c(1, i)$
(6)**For** $i = \{1 \dots n\}$
(7) **For** $j = \{1 \dots n\}$
(8) $dtw(i, j) = \|ts_1(i) - ts_2(j)\| + Min(dtw(i-1, j), dtw(i, j-1), dtw(i-1, j-1))$
(9)**Return** $dtw(n, n)$

Figure 6: Dynamic time warping algorithm

rank coefficient, $rank_2$, and calculate statistical significance, using the following standard formula:

$$p = 0.5 \cdot ErrorFunction(\frac{|z_1 - z_2|}{\sqrt{2} \cdot \sqrt{\frac{2}{N-3}}} \quad (8)$$

where N is the number of word pairs the dataset, $z_i = 0.5 \cdot \ln(\frac{1+rank_i}{1-rank_i})$, and $ErrorFunction(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ is the standard Gauss error function.

4.2 Dataset Construction Algorithm

Evaluating word relatedness is a natural ability humans have and is, therefore, considered a common baseline. To assess word relatedness, we use the WS-353 benchmark dataset, available online [14], which contains 353 word pairs. Each pair was judged, on average, by 13-16 human annotators. This dataset, to the best of our knowledge, is the largest publicly available collection of this kind, which most prior works [16, 37, 36, 35] use in their evaluation.

As an effort to provide additional evaluation data in this problem domain, we created a new dataset[1] to further evaluate our results upon. We present a principled method to create additional datasets, as opposed to the WS-353 benchmark where the word pairs were extracted manually. We propose to draw the word pairs from words which frequently occur together in large text domains. The relatedness of these pairs of words is then evaluated using human annotators, as done in the WS-353 dataset.

Selecting word pairs to evaluate: To create a balanced dataset of both related words and unrelated words, we applied the following procedure: Let W be a set of all words in the New York Times news articles. As we wish to compare between entities, we intersect this collection with entities extracted from DBpedia. We further proceed with removing stop words and rare words (words appearing less than 1000 over the entire time period), and stemmed the remaining words. We annotate this collection as W'. For each word pair $(a_i, a_j) \in W' \times W'$, their point-wise mutual information (PMI) is computed over the entire set of the articles, i.e., a group G of all possible word pairs, ordered by their PMI values

$$\begin{aligned} G \quad = \quad & \{(a_1, b_1), \dots, (a_n, b_n) \,|PMI(a_i, b_j) \\ & \leq PMI(a_{i+1}, b_{j+1}), (a_i, b_j) \in W' \times W'\} \end{aligned} \quad (9)$$

where PMI is defined as:

$$PMI(a_i, a_j) = \log \frac{p(a_i, a_j)}{p(a_i)p(a_j)} \quad (10)$$

Eventually, given a pre-defined number n of desired test pairs, every $\frac{|W' \times W'|}{n}$-th pair from the G' ordering is chosen. Formally, we construct the final set

$$D = \{g_{1+i \cdot \frac{|W' \times W'|}{n}} \,|g_j \in G, j \leq |W' \times W'|\} \quad (11)$$

Intuitively, this process performs a stratified sampling, containing both frequently and infrequently co-occurring words, with decent coverage of the entire spectrum of co-occurrence values (as measured by mutual information).

Obtaining human ratings from Amazon MTurk workers: The human "ground truth" judgements were obtained by using the Amazon's Mechanical Turk workers, in batches of 50 word pairs per assignment, resulting in 280 word pairs labeled overall. Up to 30 workers per batch were assigned, with the average of 23 MTurk workers rating each word pair, on average. Ten (distinct) pairs from WS-353 dataset were injected into each batch, in order to provide a calibration baseline to discard poor-quality work. Additionally, a simple "captcha" requiring to solve a simple math problem was given to each worker. As a result, the work of the annotators with ratings that correlated less than 50% on the WS-353 subset of the batch, or those that failed the "captcha" was

[1] http://www.technion.ac.il/ kirar/Datasets.html

discarded (approximately 7% of the submitted ratings were discarded through this procedure).

5. EXPERIMENTAL RESULTS

We first report the main experimental results comparing TSA to ESA on the WS-353 and MTurk datasets described above. Then, we analyze the performance of TSA in more detail on the WS-353 dataset to gain more insights into the effects of the different system parameters.

5.1 Main Results

In this section we compare the results of TSA to known similarity measurements. The section first provides empirical evidence that temporal signals contribute to measuring semantic relatedness of words, and then we show that our representation as a vector of concepts combined with temporal data outperforms previous temporal similarity techniques.

5.1.1 TSA vs. ESA

The comparison results of TSA on the WS-353 dataset are reported in Table 1. TSA results shown in the table are computed using cross correlation with a quadratic weighted function as the distance metric between single time series.

Table 1: TSA algorithm vs. ESA (WS-353 dataset)

Algorithm	Correlation with humans
ESA-Wikipedia [16]	0.75
ESA-ODP [16]	0.65
TSA (Section 3)	**0.80**

As reported in Table 1, TSA performs significantly better compared to the ESA-Wikipedia approach, with $p < 0.05$.

We also evaluate the performance of ESA-Wikipedia and TSA, on the additional dataset we created (we refer to it as the MTurk dataset). The results are presented in Table 2. Again, TSA performs substantially better than ESA, confirming that temporal information is useful on other datasets.

Table 2: TSA algorithm vs. state-of-the-art (MTurk Dataset)

Algorithm	Correlation with humans
ESA-Wikipedia [16]	0.59
TSA (Section 3)	**0.63**

5.1.2 TSA vs Temporal Word Similarity

Some works [8] proposed measuring semantic similarity of queries through temporal correlation analysis alone – without expending to a vector of semantic concepts. We therefore compare to additional two baselines: Word-Similarity using cross correlation and Word-Similarity using DTW as the distance measurement of the time-series of the two words. The results using the WS-353 and Mturk dataset can be seen in Table 3. In both datasets TSA significantly outperformed the baselines. This suggests that temporal vector similarity combined with static similarity is essential.

Table 3: TSA algorithm vs. temporal word similarity (WS-353 dataset)

Algorithm Dataset	WS-353	MTurk
Word-Similarity (cross correlation)	0.51	0.56
Word-Similarity (DTW)	0.59	0.58
TSA (Section 3)	**0.80**	**0.63**

5.2 TSA Performance Analysis

This section analyzes the performance of TSA for varying settings to gain more insights into the advantages and limitations of the TSA method.

5.2.1 Word Frequency Effects

To further analyze the performance of our algorithm we conducted experiments to test on which type of word pairs our algorithm outperforms the state of the art to this end. We chose to focus on word frequency. We investigated whether our algorithm performs better on frequent or rare words. We measured frequency in both domains — Wikipedia and New York Times. In order to evaluate the joint frequency of a pair of words, we combine their frequency by three types of measurements: minimum frequency of the two words, average, and maximum frequency of the two words. We divide the word pairs into three buckets, each containing an equal number of data points. We compute Spearman correlation separately in each bucket.

The results for the minimum criteria for the New York Times and Wikipedia corpora are reported in Tables 4 and 5, respectively. Similar results were obtained for the average and maximum frequency measurements. The results show that TSA performs significantly better than ESA on low-frequency words. This can be attributed to the fact that ESA is based on statistical information about words and concepts, which requires sufficient number of occurrences. Low-frequency words do not have enough statistical data, hence any additional signal, such as the temporal behavior of the words, can improve the performance.

Table 4: Grouping word pairs by NYT word frequency (WS-353 dataset)

Type of Bucket	ESA Correlation with humans	TSA Correlation with humans
Low	0.73	**0.82**
Medium	0.74	0.76
High	0.76	0.79

Table 5: Grouping word pairs by Wikipedia word frequency (WS-353 dataset)

Type of Bucket	ESA Correlation with humans	TSA Correlation with humans
Low	0.72	**0.79**
Medium	0.68	0.68
High	0.78	0.81

The results on the Mturk Dataset comparing ESA-Wikipedia, and TSA are reported in Table 6. While the absolute values of the TSA and ESA correlations with humans are lower, the trend persists: TSA significantly outperforms ESA, particularly on words with low frequency. The lower absolute

values are likely due to increased level of noise in the MTurk ratings, despite performing best-of-practice filtering of poor-quality MTurk work [31], and as explained in Section 4.2.

Table 6: Grouping word pairs by Wikipedia word frequency (Mturk dataset)

Type of Bucket	ESA Correlation with humans	TSA Correlation with humans
Low	0.52	**0.61**
Medium	0.50	0.48
High	0.77	0.79

5.2.2 Size of Temporal Concept Vector

In this subsection we experiment with several sizes of the temporal concept vector in several different natural representations. In many of the folksonomy domains presented in Section 2.1, we are able to obtain only vectors of about 10 concepts (based on API limitation in Flickr and Del.icio.us), i.e., for each word we are not able to produce all the words and their co-occurrence weight, but only the word's related tags. Due to this limitation, a traditional cosine measurement cannot be computed between those partial vectors — as each vector contains different concepts. We define a *size of a concept vector* to be the number of concepts. The main advantage of the distance measurement we defined in Section 3.1 is the ability to measure distance between vectors with different concepts representation, and even vectors of different sizes. We ran the experiments on various vector sizes. We deduce from the results (as appear in Table 7) that the optimal vector size is 10. Additional improvements for larger vector sizes might be achieved with additional feature selection.

Table 7: Effect of concept vector size on performance (WS-353)

Vector Size	5	10	50	100
Correlation with humans	0.78	**0.80**	**0.80**	0.79

5.2.3 Time Series Distance Functions

In this subsection we experiment with several distance functions, that are applied during the measurement of the semantics distance of the temporal concept vectors. Cross correlation outperforms DTW in each setting, where TSA with cross-correlation performance is 0.80, and with DTW it drops to 0.74. This indicates that, for the purpose of measuring similarity of concept's vectors, correlations in time series volume are more significant than measuring general similarity in time series structure (as in DTW).

5.2.4 Temporal Weighting Functions

Several weighting functions can be applied on the words' time series to produce higher weighting to more recent correlations (as we discussed in Section 3.2.3). In this work, we define several variations for a weighing function $f(t_1, t_2)$. This function receives two time points of two time series, and is used to weigh the distance between the time-series at these points. The functions we experiment upon are:

1. Constant Weighting Function: $f(t_1, t_2) = Constant$, which weighs all time points equally.
2. Power Weighting Function: $f(t_1, t_2) = (Max(t_1, t_2))^n$ which is a power model of weight, in which volume differences in more recent time points are weighted higher based on the power of the function. We have experimented on $n = 1, 2$.
3. Exponential Weighting Function: $f(t_1, t_2) = e^{Max(t_1, t_2)}$ which is an exponential model of weight, in which volume differences in recent time points are weighted exponentially higher.

The results of the performance for the TSA algorithm (with cross correlation distance function over WS-353) are presented in Table 8. The results provide evidence for the need to weigh the recent changes in time series distance measurement higher than the ancient changes. While linear, quadratic, and exponential temporal weighting functions perform similarly, the quadratic performs best, and we use it for all the experiments described in this paper. A few examples to illustrate those changes in performance can be seen in Table 9. It is clear from the rankings presented in the table, that quadratic weighting yields more significant correlation with human ranking than the constant weighting function. The correlation of such words, such as "Mars" and "water" in 1900 should be weighted differently from the correlation they exhibit in 2008, when NASA images suggested the presence of water on Mars.

Table 8: Effect of temporal weighting function

Temporal Weighting Function	Correlation with humans
Constant	0.70
Linear	0.79
Quadratic	0.80
Exponential	0.80

Table 9: Temporal weighting influence

Word 1	Word 2	Humans Rank	TSA-Const Rank	TSA-Quadratic Rank
Mars	water	46	210	94
peace	plan	102	220	108

6. DISCUSSION

In order to gain more intuition on which cases TSA approach should be applied, we provide real examples of the strengths and weaknesses of our methods compared to the state of the art ESA method. The results are derived from the application of the TSA algorithm with cross correlation and a quadratic weighting function as the distance metric between single time series.

6.1 Strengths of TSA

Synonyms are the first type of words for which the TSA method seems to outperform the ESA method. The reason for that is that synonyms have similar patterns of occurrence over time, as writers in the news corpus tend to use them interchangeably. Therefore, the two synonyms time-series in the same corpus strongly correlate over time. On the other hand, ESA represents each word as a vector of

concepts - where the weight is the TFIDF value. For each concept's Wikipedia article the number of distinct authors is limited, and therefore, the language model, and, as a consequence, the use of different synonyms is quite limited. For this reason, the TFIDF values of the synonyms in the ESA representation tend to be quite different. A sample of those cases can be seen in Table 10, where we present for each pair of synonyms the ranking given by human judgements, the ESA rank and the TSA rank. The rankings are based on the rank of the similarity of the pair of words out of the 353 pairs in the WS-353 dataset.

Table 10: Synonyms

Word 1	Word 2	Human Rank	ESA Rank	TSA Rank
asylum	madhouse	338	61	336
coast	shore	347	232	341
boy	lad	337	198	291
problem	challenge	209	74	252

As our method also captures co-occurrences of words in a single article (as we construct time-series aggregated over all articles on a certain date), phrases can also be identified well. ESA represents each word as a vector of Wikipedia concepts, weighted by the TFIDF of the word in the concept's article. Therefore, when measuring similarity of "hundred" and "percent' the similarity score is quite low - as the words appear in different articles and acquire completely different meanings in different contexts. Therefore, word phrases like "hundred-percent" are not identified well by ESA. More of those examples are presented in Table 11.

Table 11: Word Phrases

Word 1	Word 2	Humans Rank	ESA Rank	TSA Rank
luxury	car	189	341	235
hundred	percent	247	78	166
game	series	166	276	151

Implicit relations are one of the differentiating strengths of the TSA representation and the new distance metric we presented. For example, causality relations, such as "summer causes draught", are easily detected using correlation of the words' time-series. Relations of "type-of" (such as canyon is a type of landscape) are also relations we have found to be common when TSA outperforms ESA. We attribute that to the fact that many words in Wikipedia are associated to the general concepts (in our example "landscape") and therefore, when measuring the distance between the concepts' TFIDF vectors, the relation of each sub-object (such as "canyon") declines. Table 12 presents additional examples of pairs belonging to these relations and the ranking of human judgments, ESA and TSA algorithms for the WS-353 dataset.

6.2 Limitations of TSA

Although we have seen many results in which TSA performs better than ESA, we also present in this work some examples in which TSA performs worse.

One of the strength of the algorithm sometimes also serves as its weakness. Although this phenomenon is not too common, TSA identifies very complex implicit relations, which

Table 12: Implicit Relations

Word 1	Word 2	Humans Rank	ESA Rank	TSA Rank
closet	clothes	296	180	297
summer	drought	237	86	282
disaster	area	172	44	206
cup	tableware	217	7	283
cup	liquid	146	23	173
canyon	landscape	263	131	253
tiger	jaguar	296	201	302

are not always straight forward to humans. For example, the correlation between "drink" and "car". In the news, many times alcohol drinking correlates with car accidents, however humans tend not to find them related at all. Another representative example is "psychology" and "health". These words are considered very related by humans, however no true correlation in the news was found between the two words. More information about these examples can be seen in Table 13.

Table 13: Complex Implicit Relations

Word 1	Word 2	Humans Rank	ESA Rank	TSA Rank
drink	car	50	40	203
psychology	health	239	268	107

Some problems of our representation arise from the corpus we have selected to represent the concept's temporal behavior. The corpus is, unfortunately, sparse in certain topics — mostly specific topics such as technology and science (see Table 14). Therefore, correlations between words such as "physics" and "proton" are not identified well. A possible solution for this problem would be to add other sections of the New-York-Times news (such as sports, technology and science), and weigh the words frequency by the appearance in each one of those sections (so small sections will not be "discriminated"). Considering adding additional temporal corpus like blogs, tweets and so on, might also be useful. Unfortunately, we did not have access to this kind of data at the time.

Table 14: News Corpus Bias

Word 1	Word 2	Humans Rank	ESA Rank	TSA Rank
physics	proton	306	298	31
network	hardware	316	244	94
boxing	round	271	326	106

7. RELATED WORK

Automatically estimating word similarity (WS) and semantic relatedness (SR) have been fundamental problems for decades, and have been addressed by diverse techniques in cognitive science, computational linguistics, artificial intelligence, and information retrieval. For example, in computational linguistics, applications of WS include word sense disambiguation, information retrieval, word and text clustering [7]. This section first briefly reviews previous established approaches to the WS and SR problems; we then

focus on more recent approaches that make use of of collaboratively generated content (CGC) such as Wikipedia, and finally frame our approach in the context of previous work on using temporal information for WS and other problems.

7.1 Word Similarity and Relatedness

Until recently, computing semantic relatedness of natural language texts (ranging from a single word to a document in length) required encoding vast amounts of common-sense and domain-specific world knowledge. Prior work pursued three main directions: comparing text fragments as bags of words in vector space [2], using hand-crafted lexical resources such as WordNet [13], and using Latent Semantic Analysis (LSA) [10]. The former technique is the simplest, but performs sub-optimally when the compared texts share few words, for instance, when the texts use synonyms to convey similar messages. Unfortunately, this family of techniques are not appropriate for comparing individual words.

Lexical databases such as WordNet [13] or Roget's Thesaurus [29] encode relations between words such as synonymy, hypernymy. Multiple metrics have been proposed for computing relatedness using properties of the underlying graph structure of these resources [7, 20, 3, 28, 24, 21, 17]. A serious drawback of relying on such curated lexical resources is that it requires significant expertise and effort, and consequently such resources cover only a small fragment of the language lexicon. Specifically, such resources contain few proper names, neologisms, slang, and domain-specific technical words. Furthermore, these resources have strong lexical orientation and mainly contain information about individual words but little world knowledge in general.

In contrast, LSA [10], a purely statistical technique, leverages word cooccurrence information from a large unlabeled corpus of text. LSA does not rely on any human-organized knowledge; rather, it "learns" its representation by applying Singular Value Decomposition (SVD) to the words-by-documents cooccurrence matrix. LSA is essentially a dimensionality reduction technique that identifies a number of most prominent dimensions in the data, which are assumed to correspond to "latent concepts". Meanings of words and documents are then compared in the space defined by these concepts. Latent semantic models are notoriously difficult to interpret, since the computed concepts cannot be readily mapped into natural concepts manipulated by humans. Another statistical approach is estimating semantic relatedness of words through "distributional similarity" [23, 9] - that is, the similarity of the *contexts* in which the words occur.

In this paper we deal with "semantic relatedness" rather than "semantic similarity" or "semantic distance", which are also often used in the literature. In their extensive survey of relatedness measures, Budanitsky et al [7] argued that the notion of relatedness is more general than that of similarity, as the former subsumes many different kind of specific relations, including meronymy, antonymy, functional association, and others. They further maintained that computational linguistics applications often require measures of relatedness rather than the more narrowly defined measures of similarity. For example, word sense disambiguation can use any *related* and not just *similar* words from the context.

7.2 Using Collaboratively Generated Content

Some works [30, 26] proposed to use the Web as a source of additional knowledge for measuring similarity of short text snippets. A major limitation of this technique is that it is only applicable to short texts, because sending a long text as a query to a search engine is likely to return few or even no results at all. More closely related to our work, Gabrilovich et al. [16] presented an approach to WS that relied on exploiting Wikipedia for "Explicit Semantic Analysis" or ESA, and have demonstrated high correlation with human annotators. Strube et al. [32] also used Wikipedia for computing semantic relatedness.

7.3 Exploiting Temporal Dynamics

As many datasets have important temporal dimensions (e.g., stock quotes, sensor readings, search engine query popularity), there exist numerous techniques to analyze and mine time series data. In particular, Vlachos et al. [33] and subsequent work identified similar objects based on their trajectories through time series analysis. Among many known approaches to time series similarity we consider Dynamic Time Warping (DTW) [6], which we use as one of the methods for identifying words with similar trajectories.

Gruhl et al. [18] and others [22] analyzed temporal information diffusion in blogosphere, including the temporal patterns in word popularity. Efron [11] considered term popularity in a document collection, to assign better term weights for document ranking. More similar to our work, but in the context of analyzing temporal search engine query logs (which often exhibit strong temporal regularities [4]), some work [8, 25, 39] proposed a method for detecting semantically similar queries through temporal correlation analysis. More generally, time series analysis has been used previously to detect similar topic patterns [34], among many other applications. However, to the best of our knowledge, temporal information has not yet been used to improve general word relatedness estimation. In the related context of searching evolving document collections, several prior studies focused on versioned document retrieval models, where the objective is to efficiently access previous versions of the same document [38, 19]. Elsas and Dumais [12] studied the dynamics of document content change with applications to document ranking. Research on topic detection and tracking (TDT) analyzed the evolution of stories and topics over time [1]. Gabrilovich et al. [15] studied the dynamics of information novelty in evolving news stories. Olston and Pandey [27] introduced the notion of information longevity to devise more sophisticated crawling policies.

While our work also makes use of temporally evolving statistics of a document collection, our goal is different in that we seek to identify related words based on temporal patterns, rather then improve performance on a specific application such as ranking or web crawling. Furthermore, our work presents a novel way of representing terms in a vector space of concept time series, which can then be compared with time-series similarity measurements as building blocks. Finally, we provide a ways of combining the static and temporal information for computing relatedness - resulting in a significantly more accurate estimation of relatedness than using either signal in isolation.

8. CONCLUSIONS

We proposed a novel approach to computing semantic relatedness with the aid of a large scale temporal corpus. We use the New York Times archive that spans over a large period of time, and which, to the best of our knowledge, have

not been used before in such tasks. Specifically, we introduced two innovations over the previous words' semantic relatedness methods: first, a new method, Temporal Semantic Analysis, for representing the semantics of natural language terms, and a new method for measuring semantic relatedness of terms, using this representation. The algorithm is robust in that it can be naturally tuned to assign different weights to time periods, and can be used for studying language evolution over time.

Our empirical evaluation confirms that using TSA leads to significant improvements in computing words relatedness over two large datasets. Compared with the previous state of the art, TSA yields statistically significant improvements in correlation of computed relatedness scores with human judgements.

We also provide an algorithm for the automatic construction of new datasets of measuring semantic relatedness of words, and provide additional dataset to the community for further research in the field.

We believe that more accurate identification of word relatedness provided by TSA, will enable more intelligent search, improve text classification accuracy, and enable other tasks that normally require understanding of subtle relationships between words.

9. REFERENCES

[1] James Allan. Introduction to topic detection and tracking. In *Topic detection and tracking: event-based information organization*, pages 1–16. 2002.

[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] Satanjeev Banerjee and Ted Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, pages 805–810, 2003.

[4] S.M. Beitzel, E.C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman. Temporal analysis of a very large topically categorized web query log. *JASIST*, 58(2):166–178, 2007.

[5] R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4:1–9, 1959.

[6] D. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI-94 workshop on knowledge discovery in databases*, 1994.

[7] Alexander Budanitsky and Graeme Hirst. Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47, 2006.

[8] S. Chien and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *WWW*, pages 2–11, 2005.

[9] Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1–3):43–69, 1999.

[10] Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[11] Miles Efron. Linear time series models for term weighting in information retrieval. *JASIST*, 6(7):1299–1312, 2010.

[12] J.L. Elsas and S.T. Dumais. Leveraging temporal dynamics of document content in relevance ranking. In *WSDM*, 2010.

[13] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[14] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM TOIS*, 20:116–131, 2002.

[15] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *WWW*, pages 482–490, 2004.

[16] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, pages 1606–1611, 2007.

[17] Gregory Grefenstette. SEXTANT: Exploring unexplored contexts for semantic extraction from syntactic analysis. In *ACL'92*, pages 324–326, 1992.

[18] Daniel Gruhl, Ramanathan V. Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *WWW*, 2004.

[19] Jinru He, Hao Yan, and Torsten Suel. Compact full-text indexing of versioned document collections. In *CIKM*, pages 415–424, 2009.

[20] Mario Jarmasz. Roget's thesaurus as a lexical resource for natural language processing. Master's thesis, University of Ottawa, 2003.

[21] Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *ROCLING'97*, pages 57–63, 1997.

[22] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *WWW*, 2003.

[23] Lillian Lee. Measures of distributional similarity. In *ACL*, pages 25–32, 1999.

[24] Dekang Lin. An information-theoretic definition of word similarity. In *ICML'98*, pages 296–304, 1998.

[25] Z. Vagena M. Vlachos, C. Meek and D. Gunopulos. Identifying similarities, periodicities and bursts for online search queries. In *SIGMOD*, 2004.

[26] Donald Metzler, Susan Dumais, and Christopher Meek. Similarity measures for short segments of text. In *Advances in Information Retrieval*, volume 4425, pages 16–27. 2007.

[27] Christopher Olston and Sandeep Pandey. Recrawl scheduling based on information longevity. In *WWW*, 2008.

[28] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.

[29] Peter Roget. *Roget's Thesaurus of English Words and Phrases*. Longman Group Ltd., 1852.

[30] Mehran Sahami and Timothy Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *WWW*, pages 377–386, 2006.

[31] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*.

[32] Michael Strube and Simon Paolo Ponzetto. WikiRelate! Computing semantic relatedness using Wikipedia. In *AAAI'06*, pages 1419–1424, 2006.

[33] M. Vlachos, D. Gunopoulos, and G. Kollios. Discovering similar multidimensional trajectories. In *ICDE*, 2002.

[34] X. Wang, C.X. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD*, page 793. ACM, 2007.

[35] Eric Yeh, Daniel Ramage, Christopher D. Manning, Eneko Agirre, and Aitor Soroa. Wikiwalk: Random walks on wikipedia for semantic relatedness. In *TextGraphs Workshop*, pages 41–49, 2009.

[36] T. Zesch, C.Müller, and I. Gurevych. Using wiktionary for computing semantic relatedness. In *AAAI*, 2008.

[37] Torsten Zesch and Iryna Gurevych. Wisdom of crowds versus wisdom of linguists - measuring the semantic relatedness of words. *Journal of Natural Language Engineering.*, 16(01):25–59, 2010.

[38] Jiangong Zhang and Torsten Suel. Efficient search in large textual collections with redundancy. In *WWW*, 2007.

[39] Qiankun Zhao, Steven C. H. Hoi, and Tie yan Liu. Time-dependent semantic similarity measure of queries using historical click-through data. In *WWW*, 2006.