Text Similarity using Google Tri-grams

Aminul Islam, Evangelos Milios, and Vlado Keselj

Faculty of Computer Science
Dalhousie University, Halifax, Canada
{islam,eem,vlado}@cs.dal.ca

Abstract. The purpose of this paper is to propose an unsupervised approach for measuring the similarity of texts that can compete with supervised approaches. Finding the inherent properties of similarity between texts using a corpus in the form of a word n-gram data set is competitive with other text similarity techniques in terms of performance and practicality. Experimental results on a standard data set show that the proposed unsupervised method outperforms the state-of-the-art supervised method and the improvement achieved is statistically significant at 0.05 level. The approach is language-independent; it can be applied to other languages as long as n-grams are available.

Keywords: Text Similarity, Word Similarity, Unsupervised, Google n-gram, Tri-gram

1 Introduction

A text similarity method has many applications in natural language processing and related areas such as text summarization, word sense disambiguation (WSD), information retrieval, image retrieval, text categorization, formatted documents classification. There are other areas where text similarity plays an important role. O'Shea et al. [1] applied text similarity in Conversational Agents, which are computer programs that interact with humans through natural language dialogue. Some examples in other areas include: Database schema matching [2], Health care dialogue systems [3], and Phone call routing [4].

In practice, the majority of approaches for measuring the similarity of texts are based on a conventional domain-dependent background dictionary that represents a fixed and usually static collection of words of a given language. As a result, satisfactory similarity score can only be achieved if the dictionary covers most tokens of the texts. The types or words that are used in real-world texts, especially special text corpora, are often not found in the dictionary. Corpus-based measures generally collect n-grams (usually bi-grams) and their frequencies from a corpus and then use those statistics to determine word similarity because of the lack of off-the-shelf n-grams for a wide range of collections. For example, Islam and Inkpen [5] used corpus-based word similarity to estimate text similarity where they used the British National Corpus (BNC) to generate bi-grams for the words processed by their method. As a result, the method proposed by [5] cannot be used on-line. To collect tri-grams and their frequencies from a corpus is even more computationally expensive. This motivates us to use Google tri-grams from the Google Web 1T data set [6]. Efficient use of this data set can solve the problem of having no off-the-shelf n-gram lists.

This paper seeks to advance the state-of-the-art in text similarity by using an unsupervised statistical method. Although English is the focus of this paper, our method does not depend on any specific language, and could be used with almost no change with many other languages that have enough available n-grams. The rest of this paper is organized as follows: Section 2 presents a brief overview of the related work. Our proposed method is described in Section 3. Evaluation and experimental results are discussed in Section 4. We address some contributions and future related work in Section 5.

2 Related Work

Existing work on determining text similarity is broadly categorized into three major groups: corpus-based, knowledge-based and hybrid method.

Islam and Inkpen [5] proposed a corpus-based text similarity measure as a function of string similarity, word similarity and common word order similarity. For determining word similarity, they focused on corpus-based measures because of large type coverage in corpus. They used the Second Order Co-occurrence Pointwise Mutual Information (SOC-PMI) word similarity method [7] using n-grams collected from the BNC. Having no off-the-shelf list of n-grams for the BNC means their method needs to generate it first in order to process any specific word and as a result this approach is not time efficient. Ho et al. [8] modified Islam and Inkpen's [5] corpus-based word similarity measure into a knowledge-based word similarity measure, which is then integrated with WSD.

Liu et al. [9] integrated the Dynamic Time Warping (DTW) technique into the similarity measure by taking into account the semantic information, word order and the contribution of different parts of speech in a sentence. Feng et al. [10] proposed a method to estimate the sentence similarity by considering the 'direct relevance' and 'indirect relevance' between sentences.

Mihalcea et al. [11] suggested a hybrid method for measuring the semantic similarity of texts by exploiting the information that can be drawn from the similarity of the component words. Specifically, they used two corpus-based and six knowledge-based measures of word semantic similarity, and combined the results to show how these measures can be used to derive a text-to-text similarity metric. Li et al. [12] proposed another hybrid method that derives text similarity from semantic and syntactic information contained in the compared texts. O'Shea et al. [1] used the facility of Latent Semantic Analysis (LSA) to compare search terms, which they used to compare the similarity of two sentences.

3 Proposed Method

The proposed method determines the similarity between two texts using the tri-gram word similarity. Unlike Islam and Inkpen's [5] STS model, we do not use the string similarity and the optional common-word order similarity modules because the impact of these two modules is data dependent. However, these modules can easily be added on, if required.

3.1 n-gram Word Similarity

As Google n-grams consist of uni-grams to 5-grams, we need to answer the question of what n-gram ($n \in \{1, \ldots, 5\}$) is better for word similarity task. We tried to find the answer from the literature. Kaplan [13] observed that sense resolution given one word on each side of the word is more effective than two preceding or two following. He also observed that considering two words on either side of the word was not significantly better or worse than when given the entire sentence. This supports the effectiveness of tri-grams over bi-grams, 4-grams or 5-grams, specially on the tasks more close to sense resolution or ambiguity reduction.

First, we detail the tri-gram word similarity model and then generalize the model to n-gram word similarity model. The main idea of the tri-gram similarity model is to take into account all the tri-grams that start and end with the given pair of words and then normalize their mean frequency using uni-gram frequency of each of the words as well as the most frequent uni-gram in the corpus used.

Let us define the notations that we use in this section. Let w_a and w_b be the two words for which we need to determine the semantic relatedness, C be the maximum frequency possible among all Google uni-grams, $c(w)$ be the frequency of word w in Google uni-grams, $c(w_a w_d w_b)$ be the frequency of the tri-gram $w_a w_d w_b$ in Google tri-grams, and $\min(x, y)$ be the function that returns the minimum number between x and y. Thus, if there are n_1 tri-grams that start with word w_a and end with word w_b then the summation of the frequencies of all these tri-grams is $\sum_{i=1}^{n_1} c(w_a w_i w_b)$. Assume that there are n_2 tri-grams that start with word w_b, end with word w_a and the summation of the frequencies of all these tri-grams is $\sum_{i=1}^{n_2} c(w_b w_i w_a)$. Thus, we define a function $\mu(w_a, n_1, w_b, n_2) = \frac{1}{2}(\sum_{i=1}^{n_1} c(w_a w_i w_b) + \sum_{i=1}^{n_2} c(w_b w_i w_a))$, which represents the mean frequency of n_1 tri-grams that start with word w_a and end with word w_b and n_2 tri-grams that start with word w_b and end with word w_a. Tri-gram word similarity between w_a and w_b, $\mathrm{Sim}(w_a, w_b) \in [0, 1]$ defined as:

$$\mathrm{Sim}(w_a, w_b) = \begin{cases} \frac{\log \frac{\mu(w_a, n_1, w_b, n_2) C^2}{c(w_a) c(w_b) \min(c(w_a), c(w_b))}}{-2 \times \log \frac{\min(c(w_a), c(w_b))}{C}} & \text{if } \frac{\mu(w_a, n_1, w_b, n_2) C^2}{c(w_a) c(w_b) \min(c(w_a), c(w_b))} > 1 \\ \frac{\log 1.01}{-2 \times \log \frac{\min(c(w_a), c(w_b))}{C}} & \text{if } \frac{\mu(w_a, n_1, w_b, n_2) C^2}{c(w_a) c(w_b) \min(c(w_a), c(w_b))} <= 1 \\ 0 & \text{if } \mu(w_a, n_1, w_b, n_2) = 0 \end{cases} \tag{1}$$

The intuition of (1) is to consider the frequencies of all the tri-grams that start and end with the given pair of words with respect to the uni-gram frequencies of the pair. The only change that we need to adapt this tri-gram word similarity model to n-gram model is to use the appropriate $c(n\text{-grams})$ function. For example, in tri-grams model, $c(n\text{-grams}) = c(w_a w_d w_b)$, where $w_a w_d w_b$

is a tri-gram. Similarly, for 4-grams model, $c(n\text{-grams}) = c(w_a w_d w_e w_b)$, where $w_a w_d w_e w_b$ is a 4-gram.

3.2 Overall Text Similarity

The main idea is to find for each word in the shorter text, some most similar matchings at the word level, in the longer text. Islam and Inkpen's [5] text similarity model used one to one mapping, thus left a space to miss some significant associations. Our proposed method consists of the following five steps:

Step 1: After preprocessing (i.e., removing special characters, punctuations and stop words) we assume that the two input texts $P = \{p_1, p_2 \ldots, p_m\}$ and $R = \{r_1, r_2 \ldots, r_n\}$ have m and n tokens, respectively, and $n \geq m$. Otherwise, we switch P and R.

Step 2: We count the number of p_i's (say, δ) for which $p_i = r_j$, for all $p \in P$ and for all $r \in R$. That is, there are δ tokens in P that exactly match with R, where $\delta \leq m$. We remove all δ tokens from both of P and R. So, $P = \{p_1, p_2 \ldots, p_{m-\delta}\}$ and $R = \{r_1, r_2 \ldots, r_{n-\delta}\}$. If all the terms match, $m - \delta = 0$, we go to step 5.

Step 3: We construct a $(m-\delta) \times (n-\delta)$ 'semantic similarity matrix' (say, $M = (\alpha_{ij})_{(m-\delta)\times(n-\delta)}$) using the following process: We put α_{ij} ($\alpha_{ij} \leftarrow \mathrm{Sim}(p_i, r_j)$ using (1)) in row i and column j position of the matrix for all $i = 1 \ldots m - \delta$ and $j = 1 \ldots n - \delta$.

$$M = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1j} & \cdots & \alpha_{1(n-\delta)} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2j} & \cdots & \alpha_{2(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \cdots & \alpha_{ij} & \cdots & \alpha_{i(n-\delta)} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{(m-\delta)1} & \alpha_{(m-\delta)2} & \cdots & \alpha_{(m-\delta)j} & \cdots & \alpha_{(m-\delta)(n-\delta)} \end{pmatrix}$$

Step 4: We set notation for two known functions mean (μ) and standard deviation (σ) considering a set of x numbers, $\{a_1, \ldots, a_x\}$ as:

$$\mu(\{a_1, \ldots, a_x\}) = \frac{1}{x} \sum_{i=1}^{x} a_i$$

$$\sigma(\{a_1, \ldots, a_x\}) = \sqrt{\frac{1}{x} \sum_{i=1}^{x} (a_i - \mu(\{a_1, \ldots, a_x\}))^2}$$

For each row in M, we do the following:
Find the set of elements for any row, i, such that each element in the set is larger than the summation of the mean and standard deviation of that row. The idea is to take into account some most similar matchings unlike in other methods which consider only a single matching per word. If there are y_i such elements in the set then we can write that set, A_i, in set-builder notation as:

$$A_i = \{\alpha_{ij} : \alpha_{ij} \in \{\alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{i(n-\delta)}\}, \quad \alpha_{ij} > \mu(\{\alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{i(n-\delta)}\}) + \sigma(\{\alpha_{i1}, \ldots, \alpha_{ij}, \ldots, \alpha_{i(n-\delta)}\})\}$$

The mean of these y_i elements is $\mu(A_i)$. The summation of the means of all the $m - \delta$ rows in M is $\sum_{i=1}^{m-\delta} \mu(A_i)$.

Step 5: We add δ to $\sum_{i=1}^{m-\delta} \mu(A_i)$ and scale this total score by the reciprocal harmonic mean of m and n to obtain a normalized similarity score between 0 and 1, inclusively:

$$S(P,R) = \frac{(\delta + \sum_{i=1}^{m-\delta} \mu(A_i)) \times (m+n)}{2mn} \tag{2}$$

4 Evaluation and Experimental Results

In order to evaluate our text similarity measure, we compute the similarity score for 30 sentence pairs from [12] and find the correlation with human judges in order to compare with [12, 9, 10, 1, 5, 8] who also use the same 30 sentence pairs and find the correlation with human judges. The detailed procedure of this data set preparation is in [12]. Table 1 shows that our proposed text similarity measure achieves a high Pearson correlation coefficient of 0.916 with the mean human similarity ratings, whereas Ho et al.'s [8] similarity measure achieves 0.895. The improvement achieved is statistically significant at 0.05 level. The best participant obtained a correlation of 0.921 and the worst 0.594 with the average of the human judges that is used as expected solution to the task. Li et al. [12] calculated the correlation coefficient for the judgments of each participant against the rest of the group and then took the mean to determine the mean of all participants which is 0.825. Islam and Inkpen's [5] STS model using our proposed tri-gram word similarity achieves a Pearson correlation coefficient of 0.894 with the mean human similarity ratings.

Table 1. Similarity correlations

Name of the Measure	**Correlation**
Worst Human Participant	0.594
Mean of all Human Participants	0.825
Li et al.	0.816
Liu et al.	0.841
Feng et al.	0.756
O'Shea et al.	0.838
Islam et al. (STS)	0.853
Ho et al. (SPD-STS)	0.895
Our Method	0.916
Best Human Participant	0.921

5 Conclusion

The proposed unsupervised text similarity method achieves a good Pearson correlation coefficient for 30 sentence pairs data set and outperforms the results obtained by [8] (the improvement is statistically significant). The performance of our method is very close to that of best human participant. Our method is general enough to incorporate Islam and Inkpen's [5] string similarity and common-word order similarity module, if required. We could decide whether string similarity or common-word order similarity is required or not for a specific data set, only if we could use some training data on that data set, in which case the approach would no longer be unsupervised. In the future, we would like to test our text similarity method for long documents and in other possible applications, some of which are already mentioned in the introductory section.

References

1. O'Shea, J., Bandar, Z., Crockett, K., McLean, D.: A comparative study of two short text semantic similarity measures. In: Proceedings of the 2nd KES International conference on Agent and multi-agent systems: technologies and applications. KES-AMSTA'08, Berlin, Heidelberg, Springer-Verlag (2008) 172–181
2. Islam, A., Inkpen, D., Kiringa, I.: Applications of corpus-based semantic similarity and word segmentation to database schema matching. The VLDB Journal **17**(5) (2008) 1293–1320
3. Bickmore, T., Giorgino, T.: Health dialog systems for patients and consumers. J. of Biomedical Informatics **39** (October 2006) 556–571
4. Gorin, A.L., Riccardi, G., Wright, J.H.: How may I help you? Speech Communication **23**(1-2) (1997) 113–127
5. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discov. Data **2** (July 2008) 10:1–10:25
6. Brants, T., Franz, A.: Web 1T 5-gram corpus version 1.1. Technical report, Google Research (2006)
7. Islam, A., Inkpen, D.: Second order co-occurrence PMI for determining the semantic similarity of words. In: Proceedings of the International Conference on Language Resources and Evaluation, Genoa, Italy (May 2006) 1033–1038
8. Ho, C., Murad, M.A.A., Kadir, R.A., Doraisamy, S.C.: Word sense disambiguation-based sentence similarity. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters. COLING '10, Stroudsburg, PA, USA, Association for Computational Linguistics (2010) 418–426
9. Liu, X., Zhou, Y., Zheng, R.: Sentence similarity based on dynamic time warping. In: Proceedings of the International Conference on Semantic Computing, Washington, DC, USA, IEEE Computer Society (2007) 250–256
10. Feng, J., Zhou, Y.M., Martin, T.: Sentence similarity based on relevance. In Magdalena, L., Ojeda-Aciego, M., Verdegay, J., eds.: IPMU. (2008) 832–839
11. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the American Association for Artificial Intelligence, Boston (2006)
12. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. on Knowl. and Data Eng. **18** (August 2006) 1138–1150
13. Kaplan, A.: An experimental study of ambiguity and context. (November 1950) Published as Kaplan, Abraham (1955), An experimental study of ambiguity and context, Mechanical Translation, 2(2), 39-46.