Using Linguistic Knowledge To Determine Sentence Level Semantic Similarity

Ruchir Patel
Qiang Liu

1. Introduction

Finding similarity between two sentences is the natural progression from finding similarity between two words. Humans are pretty good at it, partly because we created the language. Finding similarity between two sentences has many applications. In world wide web, it can be used to to retrieve most relevant documents. It can be used to extract knowledge from textual database [1]. It can be evaluated to judge machine translation. A bot can use similarity between incoming text and pre existing questions to take the used on most effective conversational route.

Documents have a general subject where sentences convey very specific information hence the topic modelling and topicality isn't the best way to go. There are multiple types of Measures: Word Overlap Measures, TF-IDF, corpus-based methods, Linguistic Measures, and Machine Translation Metrics [2] [4] [5].

We show the current Linguistic Measures to find similarity between two sentences and propose a new method which outperforms the old one on Microsoft Research Paraphrase Corpus (MSRP). [6]

2. Related Work

Previous works have evaluated the performance of simplistic approach such as word-overlap, TF-IDF and document fingerprinting to statistical translation models to identify topically related sentences [15]. Lexical matching and language modelling has also been computed to find relevant sentences [16]. However, we are not focusing on topical relevance. We are focusing on finding similarity between two sentences, and try to capture what it is trying to convey. To find similarity between arbitrary sentences, some Machine Translation methods have been directly applied as well [5].

2.1 Machine Translation Evaluation Methods

2.1.1 WER

Word Error Rate (WER) [14] measures number of edit operations required to transform one sentence into another

2.1.2 PER

Position-independent word error rate (PER) [13] is WER but the word order is not taken into account

2.1.3 BLEU

Bilingual Evaluation Understudy (BLEU) [12] is geometric mean of n-gram precision.

3. Linguistic Measures

3.1 Sentence Semantic Similarity Measure

Li et al. [3] proposed a vector based approach which measures semantic sentence similarity. Each sentence is a sequence of word which carries useful information.

Here is how method proposed by Li et al. [3] works, with an example.

Given two sentences S1 and S2, we extract all words T1 and T2. A join set T is made T = T1 union T2 of length *m* as follows.

Sentence 1 = A quick brown dog jumps over the lazy dog.
Sentence 2 = A fast brown fox jumps over the lazy dog.

T1 = { 'a', 'quick', 'brown', 'dog', 'jumps', 'over', 'the', 'lazy', 'fox' }
T2 = { 'a', 'fast', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog' }

T = { 'a', 'quick', 'brown', 'dog', 'jumps', 'over', 'the', 'lazy', 'fox', 'fast' }

A lexical semantic vector, known as S is derived from the joint set T for each sentence. The entry value of this vector (of length *m)* is determined by the similarity of corresponding word in T to any word in sentence.

T = { w_1, w_2, … w_m }

We get two vectors s1 and s2 of length *m* each,

for example, for sentence T1,

case 1: For each w_i in T, if w_i appears in the sentence T1, s1_i is set to 1.
case 2: if w_i not in T1, a similarity score between w_i and each word in T1 computed. if most similar word in T1 to w_i has similarity score > threshold, s1_i is set to that score, otherwise 0.

In this paper, for now, we ignore the information content of the word i.e. probability of that word in corpus.

We, in this paper, used word2vec [7] to find similarity between two words which is a very powerful approach. Li et al. [3] used a WordNet [10] based algorithm exploiting the structure of lexical knowledge base to find similarity between to words.

s1 = [1, 1, 1, 1, 1, 1, 1, 1, 1, 0.776]
s2 = [1, 0.776, 1, 1, 1, 1, 1, 1, 1, 1]

The S_{ssv} semantic similarity between two sentences is defined as cosine coefficient of these two vectors.

3.2 Word Order Similarity

It is not possible to discriminate sentences which have a common Bag of Words. Syntactical information is the key in order to do so.

A vector R, called word order vector is created to capture syntactical information.

We get two vectors r1 and r2 of length *m* each,

for example, for sentence T1,

case 1: For each w_i in T, if w_i appears in sentence T1, r1_i is set to corresponding index number in T1.
case 2: if w_i is not in T1, a similarity score between w_i and each word in T1 is computed. If the most similar word in T1 to w_i has word similarity score > threshold, index of that word is assigned to w_i. Otherwise 0.

r1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 2]
r2 = [1, 2, 3, 9, 5, 6, 7, 8, 4, 2]

Similarity S_{wo} is measured by $1 - (\| r1 - r2 \| / \| r1 + r2 \|)$, which is by normalising the difference of word order.

3.3 Overall Similarity

A Linear combination of semantic and syntactical similarity score is calculated by Li et al. [3] to judge overall similarity.

$S = \delta * S_{ssv} + (1 - \delta) * S_{wo}$

Where $\delta <= 1$ decided the contribution of semantic and syntactical information. Empirical data shows that this algorithm performs best when more weight is given to semantic score than syntactical score [8] [9] ($\delta = 0.8$) [2].

We have used $\delta = 0.8$ while testing overall similarity in results.

3.4 Dependency Based Semantic Similarity Measure

We propose a new method which is an extension to method described in 3.1 Sentence Semantic Similarity Measure. Instead of building a 1D vector of length *m*, we build a 2D matrix of *m X m.* We used spacy [11] dependency parser to fill this matrix.

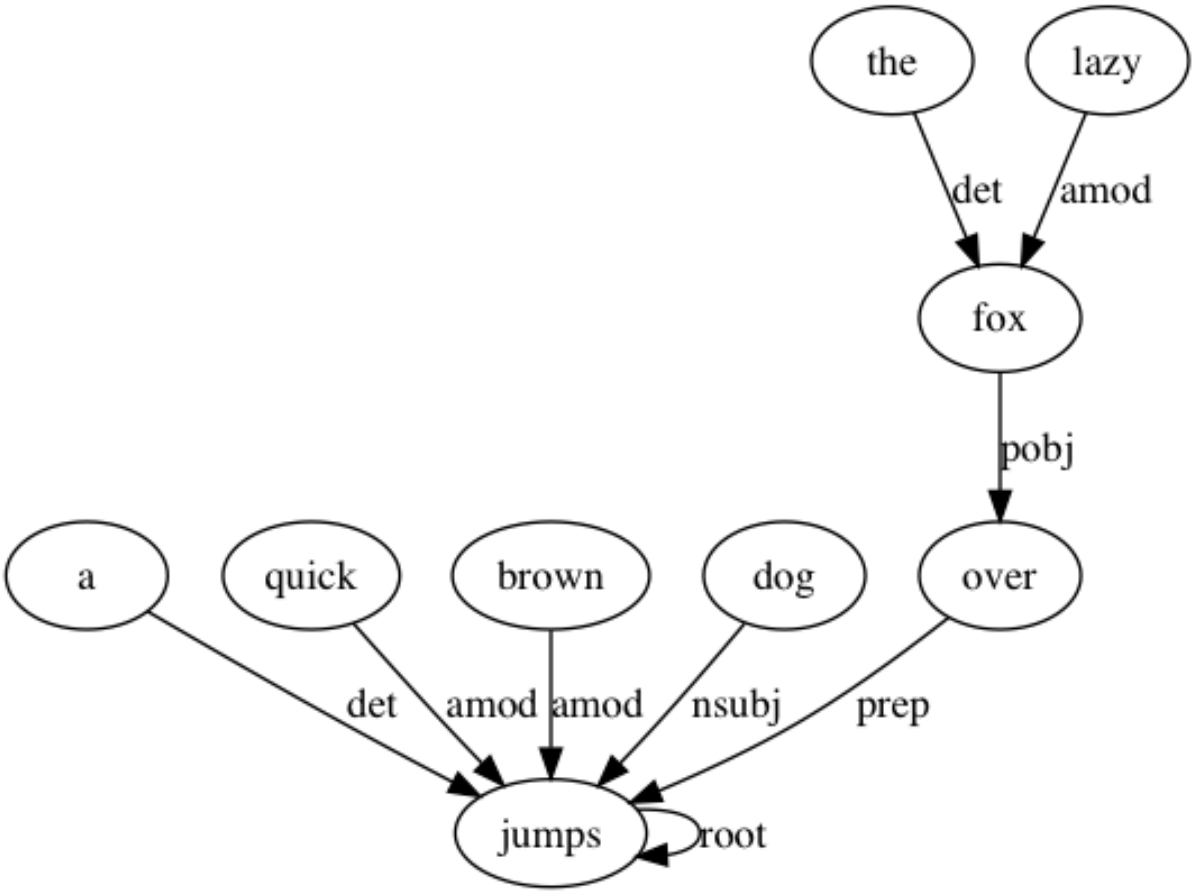
Consider following pair of sentences.

Pair 1
Sentence 1 = A quick brown dog jumps over the lazy dog.
Sentence 2 = A fast brown fox jumps over the lazy dog.

Pair 2
Sentence 1 = A quick brown dog jumps over the lazy dog.
Sentence 2 = what jumps over the lazy fox is a quick brown dog.

Sentences in Pair 2 conveys the same information. Sentences in Pair 2 are more closer than sentences in Pair 1. We want to capture this.

We get triples from dependency parser as follows: A quick brown dog jumps over the lazy dog.



We build two matrix q1 and q2 each of dimension *m X m* for each sentence. T1, T2, T, and word similarity are as described in 3.1. We have a weight penalty of 4.

For T1,

Case 1: For each w_i in T, if w_i appears in T1, q1[i][i] = 4
Case 2: if w_i not in T1, a similarity score between w_i and each word in T1 computed. if most similar word in T1 to w_i has similarity score > threshold, q1[i][i] is set to that score*4, otherwise 0.
Case 3: For every dependency, i.e. for every edge in the graph above, get the corresponding two nodes. and mark q1[x][y] = 1.

for Pair 1: T = ['a', 'quick', 'brown', 'dog', 'jumps', 'over', 'the', 'lazy', 'fox', ‘fast']

q1 matrix will be built as follows

	A	Quick	Brown	Dog	Jumps	Over	The	Lazy	Fox	Fast
A	4	0	0	0	1	0	0	0	0	0
Quick	0	4	0	0	1	0	0	0	0	0
Brown	0	0	4	0	1	0	0	0	0	0
Dog	0	0	0	4	1	0	0	0	0	0
Jumps	1	1	1	1	1	1	0	0	0	0
Over	0	0	0	0	1	4	0	0	1	0
The	0	0	0	0	0	0	4	0	1	0
Lazy	0	0	0	0	0	0	0	4	1	0
Fox	0	0	0	0	0	1	1	1	4	0
Fast	0	0	0	0	0	0	0	0	0	3.10482

Notice, “Fast” word is not present in T1, but it is similar to word “Quick” and the similarity score between Fast and Quick is 0.7762 which is more than threshold, hence q1[Fast][Fast] = 0.7762*4

There are multiple ways to find distance between two matrix. I implemented two.

$q^* = q1 - q2$
$q^{**} = q1 + q2$.

S_{dp_nz} = 1 - number of non zero (q^*) / number of non zero (q^{**})
S_{dp_norm} = 1 - normalize (q^*) / normalize (q1) + normalize (q2)

Running all four similarity measures on these pairs of sentences reveal that only my method S_{dp_nz} captures the fact that sentences in Pair 2 are more similar than in Pair 1.

	Pair 1	Pair 2
S_{ssv}	**0.994784328776**	0.904534033733
S_{wo}	**0.787378372219**	0.525229885626
S_{dp_norm}	**0.813119975816**	0.733976394451
S_{dp_nz}	0.576923076923	**0.589285714286**

Empirically, we show that counting non zero method S_{dp_nz} performs better on MSRP than S_{dp_norm}.

4. Experimental Results

4.1 Data Set

We used Microsoft Research Paraphrase Corpus (MSRP) [6] which contains 5801 pairs constructed from internet news sources. Each pair is labeled to be either paraphrase of each other, or not. Two annotators judged the task a third one was used when the two disagreed. The two agreed on 83% of the pairs, which shows it is difficult to judge such a task. 2/3 of the pair are annotated as paraphrase. These pairs are pre divided into 4076 pairs of training set and 1725 pairs in test sets.

4.2 Classification

We used linear support vector classifier from sklearn library in Python. The training set of MSRP Corpus was used for training and test was used testing.

Following table shows the accuracy of F1 score of similarity measures.

	TP	TN	FP	FN	Precision	Recall	Accuracy	F1
S_{sv}	1054	131	447	93	0.7021985	0.9189189	0.6869565	0.7960725
S_{wo}	1082	115	463	65	0.7003236	0.9433304	0.6939130	0.8038632
S_{dp_norm}	1040	134	444	107	0.7008086	0.9067131	0.6805797	0.7905739
S_{dp_nz}	1036	190	388	111	0.7275280	0.9032258	0.7107246	0.8059120
S_{sv} + S_{wo}	1045	144	434	102	0.7065584	0.9110723	0.6892753	0.7958872
S_{dp_nz} + S_{wo}	1026	203	375	121	0.7323340	0.8945074	**0.7124637**	**0.8053375**

Our combined score S_{sv} + S_{wo} outperforms one calculated by Achananuparp et al. [2] because we have used word2vec [7] to find similarity between two words instead of wordnet.

Our proposed matrix based method with dependency parser S_{dp_nz} and S_{dp_nz} + S_{wo} out performs the vector based approach.

References

[1] J. Atkinson-Abutridy, C. Mellish, and S. Aitken, “Combining Information Extraction with Genetic Algorithms for Text Mining,” IEEE Intelligent Systems, vol. 19, no. 3, 2004.

[2] Palakorn Achananuparp, Xiaohua Hu, and Shen Xiajiong "The Evaluation of Sentence Similarity Measures" http://www.cis.drexel.edu/faculty/thu/research-papers/dawak-547.pdf

[3] Li, Y., McLean, D., Bandar, Z. A., O'Shea, J. D., and Crockett, K. (2006) Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Transactions on Knowledge and Data Engineering 18, 8, 1138-1150.

[4]. Madnani, N., Tetreault, J., and Chodorow, M. (2012). Re-examining Machine Translation Metrics for Paraphrase Identification

[5]. Wan, S., Dras, M., Dale, R., and Paris, C. (2006). Using dependency-based features to take the "para-farce" out of paraphrase

[6] Dolan, W., Quirk, C., and Brockett, C. (2004) Unsupervised construction of large paraphrase corpora: Exploiting massively parallel new sources. In Proceedings of the 20th International Conference on Computational Linguistics.

[7] T Mikolov, I Sutskever, K Chen, GS Corrado, J Dean
Advances in neural information processing systems, 3111-3119

[8] Achananuparp, P., Hu, X., Zhou, X., and Zhang, X. (2008) Utilizing Sentence Similarity and Question Type Similarity to Response to Similar Questions in Knowledge-Sharing Community. To appear in Proceedings of QAWeb 2008 Workshop, Beijing, China.

[9] 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 Proc. 19th Ann. Meeting of the Cognitive Science Soc., 412-417.

[10] G.A. Miller, “WordNet: A Lexical Database for English,” Comm. ACM, vol. 38, no. 11, pp. 39-41, 1995.

[11] https://spacy.io/docs/usage/showcase

[12] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2001.
Bleu: a Method for Automatic Evaluation of Machine Translation. IBM Research Report rc22176 (w0109022), Thomas J. Watson Research Center.

[13] C. Tillmann, S. Vogel, H. Ney, A. Zubiaga, and H. Sawaf. 1997. Accelerated dp based search for statistical translation. In Proceedings of Eurospeech-97, pages 2667–2670, Rhodes, Greece.

[14] K.Y. Su, M.W. Wu, and J.S. Chang. 1992. A new quantitative quality measure for machine transla- tion systems. In Proceedings of COLING-92, pages 433–439, Nantes, France.

[15] Metzler, D., Bernstein, Y., Croft, W., Moffat, A., and Zobel, J. (2005) Similarity measures for tracking information flow. Proceedings of CIKM, 517–524.

[16] Metzler, D., Dumais, S. T., and Meek, C. (2007) Similarity Measures for Short Segments of Text. In Proceedings of ECIR 2007, 16-27.