Comparison of Algorithmic and Human Assessments of Sentence Similarity

John G. Mersch
Department of Mathematics
Xavier University of Louisiana
New Orleans, LA 70125 USA
jmersch@xula.edu

R. Raymond Lang
Department of Computer Science
Xavier University of Louisiana
New Orleans, LA 70125 USA
rlang@xula.edu

Abstract

This paper describes a new method, based on information theory, for measuring sentence similarity. The method first computes the information content (IC) of dependency triples using corpus statistics generated by processing the Open American National Corpus (OANC) with the Stanford Parser. We define the similarity of two sentences as a function of (1) the similarity of their constituent dependency triples, and (2) the position of the triples in their respective dependency trees. We apply the algorithm to 15 pairs of sentences that were also given to human subjects to assign a similarity score. The human- and computer-generated scores are compared; the results are promising, but point to the need for further refinement.

1 Introduction

This project seeks to develop an algorithm that measures the extent to which the meanings of two given sentences overlap. Our plan is to use such an algorithm in a clustering application (Lang and Mersch, 2012).

The technique described in this paper extends previous work applying an information-theoretic definition of similarity to a number of different domains (Lin, 1998). Lin's information-theoretic definition of similarity performs as well as or better than other information-theoretic similarity metrics that leverage domain specifics (Resnik, 1995; Wu and Palmer, 1994).

The metric being proposed in this paper shares characteristics of word co-occurrence methods and descriptive feature-based methods (Li et al., 2006), in addition to using structural information provided by the Stanford Parser (Klein and Manning, 2003). We test this metric on 15 pairs of sentences, each of which was assessed for similarity by 40 fluent English speakers.

2 Background & Related Work

Methods that detect similarity of long documents often utilize co-occurring words (Salton, 1988), since similar texts share a high number of words. But this does not transfer well to short, sentence-length texts, since language allows similar meanings to be expressed using different vocabularies.

Existing text similarity measures suffer from drawbacks. Vector-based methods employ high-dimensional, sparse representations that are computationally inefficient (Landauer et al., 1998; Salton, 1988; Burgess et al., 1998). Some methods rely on extensive manual preprocessing (McClelland and Kawamoto, 1986), making them impractical for large-scale use. Still other methods suffer from domain dependency (Li et al., 2006).

Related work on text similarity may be grouped into three categories:

1. Methods based on word co-occurrence (i.e. "bag of words" methods) disregard the impact of word order on meaning (Meadow et al., 1999); thus, the two sentences:

   T_1: The cat killed the mouse.
   T_2: The mouse killed the cat.

   are regarded as identical, since they use the same words. Documents are represented as vectors in an n-dimensional space, where n is the length of a pre-compiled word list, typically in the tens or hundreds of thousands. The resulting representations are sparse and computationally inefficient (Li et al., 2006). Also, these methods often exclude function words (e.g. the, of, an, etc.) that have low relevance for similarity of long documents but convey information important for sentence similarity. These methods will not detect similarity of sentences that use different

International Joint Conference on Natural Language Processing, pages 1306–1311,
Nagoya, Japan, 14-18 October 2013.

words to convey the same meaning. However, they achieve improved results by examining word pairs instead of single words (Okazaki et al., 2003).

2. Corpus-based methods. Latent semantic analysis (LSA) constructs an occurrence count matrix where the rows represent words and the columns text units, usually paragraphs or documents. It is more suitable for longer texts than for sentences (Landauer et al., 1998). Hyperspace Analogues to Language (HAL) (Burgess et al., 1998) constructs a word co-occurrence matrix based on a moving window of a predefined width, typically 10. HAL is also more effective for longer texts than for sentences (Li et al., 2006).

3. Descriptive feature-vector methods. These methods employ pre-defined thematic features to represent a sentence as a vector of feature values, then obtain a similarity measurement through a trained classifier (Taraban and McClelland, 1988). Choosing a suitable set of features and automatically obtaining values for features pose obstacles for these methods (Islam and Inkpen, 2008).

In contrast to the above approaches, Lin (1998) proposes an information-theoretic measure of similarity. This measure is derived from assumptions about similarity rather than from a domain-specific formula. The metric can be applied to any domain with a probabilistic model. From a set of assumptions grounded in information theory, Lin proves a Similarity Theorem:

> the similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are:
>
> $$\mathrm{sim}(A, B) = \frac{\log P(\mathrm{common}(A, B))}{\log P(\mathrm{description}(A, B))}$$
>
> [...] If we know the commonality of the two objects, their similarity tells us how much more information is needed to determine what these two objects are. (Lin, 1998)

Lin applies the definition to four different domains; one of these is similarity between words according to the distribution of dependency triples extracted from a text corpus. Lin's test uses a database of 14 million dependency triples extracted from a corpus consisting of items from the *Wall Street Journal* and from the *San Jose Mercury*. He also applies it to semantic similarity in a taxonomy. Lin achieves better results than distance-based definitions of similarity; his results correlate slightly better with human judgment than measures proposed by Resnik (1995) and by Wu and Palmer (1994). To illustrate the domain independence of his measure, Lin also applies it to the domain of ordinal values.

3 Approach

The Stanford Parser (de Marneffe et al., 2006) was applied to the Open American National Corpus (Ide and Suderman, 2004) to produce a database containing the counts of occurrences of all the dependency triples, which are of the form <*role, governor, dependent*>, appearing in the corpus. Cover and Thomas (2006) define the information content of a proposition as the negative logarithm of its probability. We use this definition to compute the information content of the triples occurring in the corpus. Given a dependency triple, we define two predicates:

- A governor-position predicate substitutes a variable for the governor in the triple.
- A dependent-position predicate substitutes a variable for the dependent in the triple.

For example,

t_1: <dobj, grow, tomato>

is one of the dependency triples occurring in the sentence:

s_1: The gardener has grown tomatoes.

The governor-position predicate corresponding to t_1 is:

p_1: <dobj, _G, tomato>

which binds to all occurrences of "tomato" as a direct object; the dependent-position predicate is:

p_2: <dobj, grow, _D>

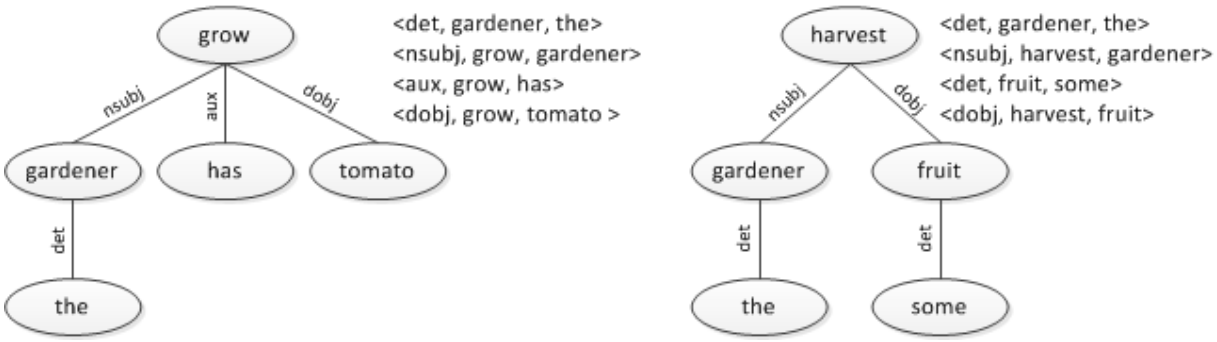

Figure 1: Dependency trees and dependency triples for s_1 and s_2

which binds to all occurrences of "grow" as a transitive verb. The information content of t_1 is computed from the number of occurrences of instantiations of its dependent-position predicate. In general, let A be the number of occurrences of <r, g, d> and let B be the number of occurrences of instantiations of <r, g, _D>. The information content of <r, g, d> is defined by:

$$\mathrm{IC}(< \mathrm{r}, \mathrm{g}, \mathrm{d} >) = -\log \frac{A}{B}$$

Next, we define similarity of two dependency triples using Lin's information-theoretic definition of similarity. The definition is explained by an example computing the similarity between the following:

t_1: <dobj, grow, tomato>
t_2: <dobj, harvest, fruit>

where t_2 is a triple from the sentence:

s_2: The gardener harvested some fruit.

The predicates p_1 and p_2 (above) are formed from t_1; from t_2, we form the predicates:

p_3: <dobj, _G, fruit>
p_4: <dobj, harvest, _D>

For each of these of these predicates, we form the set of all instantiations, $M(p_n)$. The numbers following the triples are hypothetical values for $IC(t_n)$:

$M(p_1)$: {<dobj, grow, tomato> 1.7, <dobj, raise, tomato> 3.8, <dobj, eat, tomato> 2.4}

$M(p_2)$: {<dobj, grow, tomato> 1.7, <dobj, grow, strawberry> 2.7, <dobj, grow, beard> 5.6}

$M(p_3)$: {<dobj, grow, fruit> 3.9, <dobj, harvest, fruit> 7.2, <dobj, eat, fruit> 1.2}

$M(p_4)$: {<dobj, harvest, tomato> 8.7, <dobj, harvest, strawberry> 9.7, <dobj, harvest, fruit> 7.2}

For the two governor-position predicates, p_1 and p_3, we compute the quotient of (1) the sum of the ICs of triples in $M(p_1)$ and $M(p_3)$ that have the same word in the governor position and (2) the sum of the ICs of all the triples in $M(p_1)$ and $M(p_3)$. Triples that appear in both models are counted both times. Call this quotient S_g.

$$S_g = \frac{1.7 + 2.4 + 3.9 + 1.2}{1.7 + 3.8 + 2.4 + 3.9 + 7.2 + 1.2}$$

We form the quotient S_d similarly, using the dependent-position predicates. Finally, we define

$$\mathrm{sim}(t_1, t_2) = \alpha \cdot S_g + (1 - \alpha) \cdot S_d$$

where α is a real value between zero and one.

We extend this definition of similarity between triples to define similarity between sentences. Given two sentences, the nodes of their respective dependency trees are words and the tree edges are dependency relations. For example, the triple <dobj, grow, tomato> indicates that *grow* and *tomato* are two nodes in the dependency tree and that there is a directed edge from *grow* to *tomato* labeled *dobj*.

Given two dependency trees and two nodes, one from each of the given trees, we form a collection of pairs where the first component is a branch that has the first node in the governor position and the second component is a branch that has the second node in the governor position. The process for forming this collection is as follows:

	s_1	s_2	Survey Average	Tree Similarity
1	The cat killed the mouse.	The mouse killed the cat.	0.5	0.633
2	The man walked to the store.	The person went to the store.	3.625	0.512
3	The student killed time.	The student killed the roach.	0.35	0.134
4	The janitor cleaned the desk.	The desk was cleaned by the janitor.	4.85	0.001
5	The locksmith went to the movies.	The window was stuck shut.	0.075	0.108
6	The dog went missing for three days.	The squirrel avoided the trap.	0.075	0.131
7	The student ran out of notebook paper.	The printer ran out of paper.	1.2	0.632
8	The door is open.	The door is closed.	0.5	0.330
9	Traffic downtown is heavy.	The downtown area is crowded.	2.4	0.075
10	The secretary stopped for coffee on the way to the office.	The office worker went out for dinner after work.	0.675	0.030
11	Biologists discovered a new species of ant.	Physicists verified the existence of black holes.	0.45	0.060
12	The artist drew a picture of the landscape.	The artist sketched a picture of the landscape.	4.375	0.675
13	The bear searched for food at the picnic grounds.	The bear scavenged the park for food.	3.525	0.500
14	A college degree allows one to have a rewarding career.	A bachelor's degree is necessary to get a high paying job.	2.125	0.294
15	The train arrives at half past three.	The visitor will be in the station this afternoon.	1.125	0.093

Table 1: Sentence Pairs with human subject survey averages and tree similarity measures. Survey averages range from 0 to 5; tree similarity measures range from 0 to 1.

- The triple with the highest information content from the collection of triples that have one of the given nodes in the governor position is identified. This triple may come from either tree.
- A search is done for the most similar triple from the other dependency tree.
- The two triples just identified are matched and removed from consideration. The process repeats until all of the branches exiting from one of the nodes have been matched.

Matching triples enables the recursive comparison of nodes from different dependency trees. We define the similarity of two nodes as the weighted average of:

- the similarity of the triples matched as described above;
- the result of recursively computing similarity of matched dependents (nodes one level deeper in the dependency tree); and
- unmatched branches, defined as having a similarity of zero (The two nodes may have unequal numbers of children).

The similarity of two sentences is the similarity of their root nodes.

4 Results

We applied the algorithm to 15 pairs of sentences written for the purpose of testing the approach. We asked 40 native English speakers to rank the similarity of each pair on a scale of 0 to 5, where 0 indicates "no overlap in meaning" and 5 indicates "complete overlap in meaning." The tree similarity algorithm was applied to the sentence pairs. Table 1 shows the results (survey averages range from 0 to 5; tree similarity measures range from 0 to 1).

The two similarity measures have a correlation coefficient of 0.279; however, inter-annotator

agreement was low (Fleiss's kappa = 0.313). Pairs 7, 10, 14, and 15 had the lowest inter-annotator agreement. Without these pairs, the 11 pairs that remaine (1, 2, 3, 4, 5, 6, 8, 9, 11, 12, and 13) have kappa = 0.399 and have a correlation coefficient of 0.291 with the tree similarity algorithm. Pair 4, the active/passive switch, is incorrectly scored 0 by the algorithm, whereas the annotators were in strong agreement of a rating close to 5. Removing pair 4 from the analysis (which lowers kappa) gives a correlation coefficient of 0.618 between annotator averages and the algorithm results. These results suggest that, once the algorithm is refined to properly handle the active/passive switch, it will provide results that correlate to the judgment of native speakers.

5 Contributions & Future Work

Our approach is grounded in information theory. The representation avoids high-dimensional, sparse vectors; this allows the use of the trained database without having to condense it.

Previously Lang (2010) proposed implementing Lévi-Strauss's procedure for finding the structure of a myth (Lévi-Strauss, 1955). We plan to apply the tree similarity metric in a clustering algorithm for grouping sentences into categories corresponding to the constituent terms of his canonical formula.

References

Curt Burgess, Kay Livesay, and Kevin Lund. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes*, 25(2–3):211–257.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*. John Wiley & Sons, Hoboken, New Jersey, second edition.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 449–454, Genoa, May.

Nancy Ide and Keith Suderman. 2004. The American National Corpus first release. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1681–1684, Lisbon, May.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):10:1–10:25, July.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan, July.

Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25(2–3):259–284.

R. Raymond Lang and John G. Mersch. 2012. An experiment to determine whether clustering will reveal mythemes. In Mark A. Finlayson, editor, *Proceedings of the Third Workshop on Computational Models of Narrative*, pages 20–21, Istanbul, May.

R. Raymond Lang. 2010. Considerations in representing myths, legends, and folktales. In *Computational Models of Narrative: Papers from the AAAI Fall Symposium*, pages 29–30, Arlington, VA, November.

Claude Lévi-Strauss. 1955. The structural study of myth. *The Journal of American Folklore*, 68(270):428–444.

Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150, August.

Dekang Lin. 1998. An information-theoretic definition of similarity. In Jude Shavlik, editor, *Machine Learning: Proceedings of the Fifteenth International Conference*, pages 296–304, Madison, Wisconsin, July. Morgan Kaufmann Publishers.

J. L. McClelland and A. H. Kawamoto. 1986. Mechanisms of sentence processing: Assigning roles to constituents. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 2: Psychological and Biological Models*, pages 272–325. MIT Press, Cambridge, MA.

Charles T. Meadow, Donald H. Kraft, and Bert R. Boyce. 1999. *Text Information Retrieval Systems*. Academic Press, Orlando, FL, 2nd edition.

Naoaki Okazaki, Yutaka Matsuo, Naohiro Matsumura, and Mitsuru Ishizuka. 2003. Sentence extraction by spreading activation through sentence similarity. *IEICE Transactions on Information and Systems*, E86-D(9):1686–1694.

Philip Resnik. 1995. Disambiguating noun groupings with respect to WordNet senses. In David Yarowsky and Kenneth Chruch, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 54–68, Cambridge, Massachusetts, June.

Gerard Salton. 1988. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Roman Taraban and James L. McClelland. 1988. Constituent attachment and thematic role assignment in sentence processing: Influences of content-based expectations. *Journal of Memory and Language*, 27(6):597–632, December.

Zhibiao Wu and Martha Palmer. 1994. Verb semantics and lexical selection. In *32nd Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference*, pages 133–138, Las Cruces, New Mexico, June. Association for Computational Linguistics.