An Information-Theoretic Sentence Similarity Metric

John G. Mersch
Department of Mathematics
Xavier University of Louisiana
jmersch@xula.edu

R. Raymond Lang
Department of Computer Science
Xavier University of Louisiana
rlang@xula.edu

Abstract

We describe an information theoretic-based metric for sentence similarity. The method uses 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 compare results of the algorithm to human judgments of similarity of 1725 sentence pairs.

Motivation

In "The Structural Study of Myth," Claude Lévi-Strauss proposes a method for analyzing myth that aims to elicit the existential dilemma that a myth addresses. In the process of presenting his method, Lévi-Strauss introduced a formula that has since come to be known as the canonical formula of myth; this formula has guided and informed subsequent scholarship in anthropology, linguistics, and computer science, and it is this same formula that is the object of study in our project. Since Lévi-Strauss ends his paper with a discussion of the technology (not available in the 1950s) that would be required to fully implement his method (Lévi-Strauss 1955, p. 443, paragraph 8.0), we propose to return to Lévi-Strauss' structural analysis of myth, bringing to bear the contemporary computing power that his method prefigures.

Lévi-Strauss's method for analyzing myth begins by identifying the gross constituent units, which he equates with the predicate relations, from which the myth is composed. The core hypothesis of his argument is that "the true constituent units of a myth are not the isolated relations but *bundles of such relations*" [emphasis in the original](Lévi-Strauss 1955, p. 431). The next step in the analysis is to determine for each predicate relation which bundle with which it is associated. However, this process is circular since the bundles are not known until the component relations have been properly assigned to the bundles. The criteria for determining when the bundles have been properly identified are "the principles which serve as a basis for any kind of structural analysis: economy of explanation; unity of solution; and ability to reconstruct the whole from a fragment, as well as further stages from previous ones" (Lévi-Strauss 1955, p. 431). Our project investigates the feasibility of algorithmically determining the bundles of predicate relations associated with a given myth without resorting to expert intuition.

The canonical formula, which Lévi-Strauss writes as $f_x(a) : f_y(b) \approx f_x(b) : f_{a^{-1}}(y)$, represents the abstract relationship among bundles. The formula specifies that (a) a narrative presents two pair of contrasts, one between functions and one between objects, and (b) the functions and objects combine to form four terms, the fourth of which expresses an object turned function over a function turned object. The canonical formula does not specify the arrangement of the events that make up the story. Instead, the terms of the formula define a partition of the story's elements unrelated to their position in the story.

Lévi-Strauss hints at the use of contemporary computers as an aid in discovering the bundles associated with a myth when he writes:

> A variant of average length needs several hundred cards to be properly analyzed. To discover a suitable pattern of rows and columns for those cards, special devices are needed, consisting of vertical boards about two meters long and one and one-half meters high, where cards can be pigeon-holed and moved at will; .... [A]s soon as the frame of reference becomes multi-dimensional the board-system has to be replaced by perforated cards which in turn require I.B.M. equipment, etc. (Lévi-Strauss 1955, p. 443)

Our project aims to implement Lévi-Strauss's method using machines he envisioned, but did not have in the 1950s (Lang and Mersch 2012). In this article, we present an algorithm to measure the similarity of two given sentences. The technique described in this paper extends previous work by Lin (Lin 1998) applying an information-theoretic definition of similarity to various domains. Lin's information-theoretic definition of similarity outperforms other information-theoretic similarity metrics that leverage domain specifics (Resnik 1995) (Wu and Palmer 1994). Following development and verification of this metric, we will use it in a clustering algorithm where the objects being clustered are a myth's gross constituent units. Given that Lévi-

Strauss's method amounts to a clustering application, our proposal is a suitable computational model of his approach.

Our proposed metric 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 compare this metric against human judgments of the equivalence of 1725 pairs of sentences (Quirk, Brockett, and Dolan 2004).

Background and Related Work

To arrange gross constituent units into bundles, there must be a measure for the similarity of any two given sentences. Existing methods for measuring similarity of long documents are unsuitable at the sentence level. Methods that utilize co-occurring words disregard the impact of word order on meaning (Meadow, Kraft, and Boyce 1999); for example, the two sentences, "The cat killed the mouse." and "The mouse killed the cat." are regarded as identical, since they use the same words.

Vector-based methods employ high-dimensional, sparse representations that are computationally inefficient (Salton 1989). Corpus-based methods such as latent semantic analysis (LSA) (Landauer, Foltz, and Laham 1998) and hyperspace analogues to language (HAL) (Burgess, Livesay, and Lund 1998) are more effective for longer texts than for sentences (Li et al. 2006). Descriptive feature-vector methods employ pre-defined thematic features to represent a sentence as a vector of feature values. Choosing a suitable set of features and automatically obtaining values for features pose obstacles for these methods (Islam and Inkpen 2008).

Our method uses Lin's information-theoretic measure of similarity. This measure is derived from assumptions about similarity rather than from a domain-specific formula and is applicable to any domain with a probabilistic model. From these assumptions Lin proved the following Similarity Theorem:

> [T]he 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:
>
> $$\mathrm{sim}(A,B) = \frac{\log \Pr(\mathrm{common}(A,B))}{\log \Pr(\mathrm{description}(A,B))}$$
>
> [...] 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 achieves better results than distance-based definitions of similarity; his results correlate slightly better with human judgment than sentence similarity measures proposed in (Resnik 1995) and in (Wu and Palmer 1994).

Approach

The Stanford Parser (de Marneffe, MacCartney, and Manning 2006) is applied to the Open American National Corpus (Ide and Suderman 2004) producing a database of counts of occurrences of typed dependency triples of the form [role, governor, dependent], appearing in the corpus. A proposition's information content is defined as the negative logarithm of its probability. We use this definition to compute the information content of the triples in the corpus. Given a dependency triple, we define two predicates, (1) a governor-position predicate substitutes a variable for the governor in the triple, and (2) a dependent-position predicate substitutes a variable for the dependent in the triple.

For example,

t_1: [dobj, build, desk]

is one of the dependency triples occurring in the sentence:

s_1: The carpenter has built the desk.

The governor-position predicate corresponding to t_1 is:

p_1: [dobj, G, desk]

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

p_2: [dobj, build, D]

which binds to all occurrences of "build" as a transitive verb. Each predicate has an information content based on its governor-position predicate, $\mathrm{IC_G}$, and another based on the dependent-position predicate, $\mathrm{IC_D}$. $\mathrm{IC_G}$ of t_1 is computed from the number of occurrences of instantiations of its governor-position predicate. 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 governor-position information content of [r, g, d] is defined by:

$$\mathrm{IC_G}\left([\mathrm{r,g,d}]\right) = -\ln\left(\frac{\mathrm{A}}{\mathrm{B}}\right)$$

$\mathrm{IC_D}$ is defined similarly, using the dependent-position predicate rather than the governor-position predicate. Next, we define similarity of two dependency triples using Lin's definition. The definition is explained by an example computing the similarity between the following:

t_1: [dobj, build, desk]
t_2: [dobj, buy, wood]

where t_2 is a triple from the sentence:

s_2: The carpenter bought some wood.

Figure 1 shows s_1 and s_2 with their respective parse trees and dependency triples. The predicates p_1 and p_2 (above) are formed from t_1; from t_2, we form the predicates:

p_3: [dobj, G, wood]
p_4: [dobj, buy, D]

For each of these of these predicates, we form the set of all instantiations, M(p_n), called the model of p_n. The numbers following the triples are hypothetical values for $\mathrm{IC_G}(t_n)$:

M(p_1) : {[dobj, build, desk] $\rightarrow$ 1.7, [dobj, paint, desk] $\rightarrow$3.8, [dobj, buy, desk] $\rightarrow$8.7}

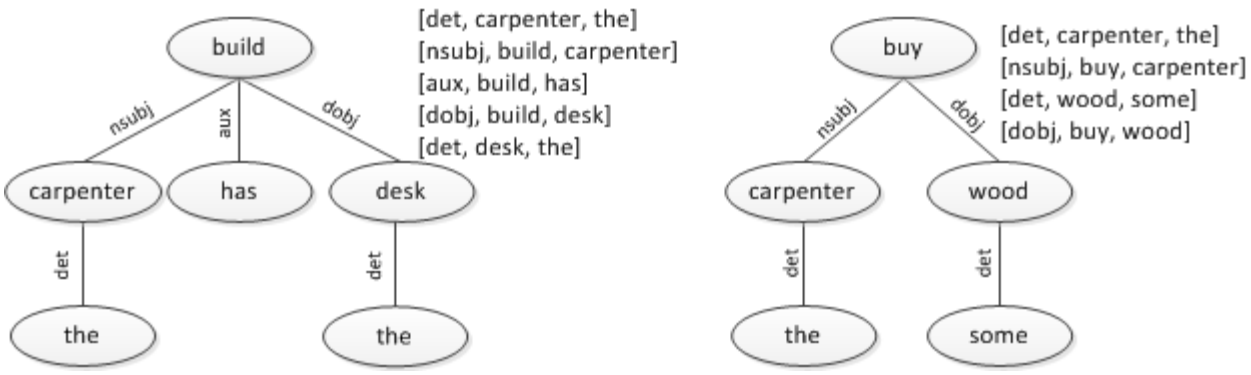

Figure 1: Dependency trees and dependency triples for s_1 and s_2

M(p_2): {[dobj, build, desk]→ 1.7, [dobj, build, chair] →2.7, [dobj, build, house]→ 5.6}

M(p_3): {[dobj, chop, wood] →3.9, [dobj, buy, wood] →7.2, [dobj, paint, wood] →1.2}

M(p_4): {[dobj, buy, desk] →8.7, [dobj, buy, wood] →7.2, [dobj, buy, chair] →7.2}

For the two governor-position predicates, p_1 and p_3, we compute the quotient of (1) the sum of the $\mathrm{IC_G}$'s 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 ICG's of all the triples in the disjoint union of M(p_1) and M(p_3). Call this quotient S_g.

$$S_g = \frac{3.8 + 8.7 + 7.2 + 1.2}{1.7 + 3.8 + 8.7 + 3.9 + 7.2 + 1.2}$$

We form the quotient S_d similarly, using the dependent-position information content. 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, build, desk] indicates that build and desk are two nodes in the dependency tree and that there is a directed edge from build to desk labeled dobj.

Given two dependency trees and two nodes, one from each of the given trees, we form a collection of pairs as follows:

- 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.

Evaluation

We evaluated the algorithm in two ways. For the first evaluation, we applied the algorithm to 15 pairs of sentences written for the purpose of testing the approach. We asked 40 fluent 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 are scaled from 0 to 1 to match the scale of the tree similarity algorithm).

The two similarity measures have a correlation coefficient of 0.355; 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 remain (1, 2, 3, 4, 5, 6, 8, 9, 11, 12, and 13) have kappa = 0.399 and have a correlation coefficient of 0.383 with the tree similarity algorithm. Pair 4, the active/passive switch, is incorrectly scored 0.262 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 0.518 between annotator averages and the algorithm results.

The second evaluation was conducted using a subset of the Microsoft Research Paraphrase Corpus (Quirk, Brockett, and Dolan 2004). This entire corpus is 5801 pairs of sentences. For each pair, human judges made a binary decision whether the two sentences were paraphrases of each other. The corpus is divided into a training set (4076 pairs) and a testing set (1725 pairs). We applied our algorithm to the pairs in the testing set, generating a similarity score for each pair. In order to apply our algorithm to the task of paraphrase classification, we need to identify a value above which pairs

s_1	s_2	Survey Averages	Tree Similarity Metric
The cat killed the mouse.	The mouse killed the cat.	0.100	0.593
The man walked to the store.	The person went to the store.	0.725	0.521
The student killed time.	The student killed the roach.	0.070	0.500
The janitor cleaned the desk.	The desk was cleaned by the janitor.	0.970	0.262
The locksmith went to the movies.	The window was stuck shut.	0.015	0.085
The dog went missing for three days.	The squirrel avoided the trap.	0.015	0.133
The student ran out of notebook paper.	The printer ran out of paper.	0.240	0.638
The door is open.	The door is closed.	0.100	0.334
Traffic downtown is heavy.	The downtown area is crowded.	0.480	0.077
The secretary stopped for coffee on the way to the office.	The office worker went out for dinner after work.	0.135	0.104
Biologists discovered a new species of ant.	Physicists verified the existence of black holes.	0.090	0.059
The artist drew a picture of the landscape.	The artist sketched a picture of the landscape.	0.875	0.678
The bear searched for food at the picnic grounds.	The bear scavenged the park for food.	0.705	0.464
A college degree allows one to have a rewarding career.	A bachelor's degree is necessary to get a high paying job.	0.425	0.294
The train arrives at half past three.	The visitor will be in the station this afternoon.	0.225	0.180

Table 1: Sentence pairs with human subject survey averages and tree similarity measures

are judged to be paraphrases. We found this value by maximizing the Matthews correlation coefficient between the human judgments and our results: by these means, we achieved a Matthews correlation coefficient of 0.193 between annotator judgments and algorithmic judgment.

Conclusions and Future Work

Previously Lang (Lang 2010) proposed implementing Lévi-Strauss's procedure for finding the structure of a myth (Lévi-Strauss 1955). The project's current objective is the development of a sentence similarity metric for use in grouping gross constituent units (predicate relations) into the bundles corresponding to the terms of the canonical formula. Our sentence similarity metric 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. In future work, we will use the tree similarity metric in a clustering algorithm for grouping sentences into categories corresponding to the constituent terms of his canonical formula.

References

Burgess, C.; Livesay, K.; and Lund, K. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes* 25(2-3):211–257.

de Marneffe, M.-C.; MacCartney, B.; and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Language Resources and Evaluation Conference (LREC)*.

Ide, N., and Suderman, K. 2004. The American National Corpus first release. In *Proceedings of the Fourth International Language Resources and Evaluation Conference (LREC)*.

Islam, A., and Inkpen, D. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 2(2).

Klein, D., and Manning, C. D. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual meeting of the Association for Comutational Linguistics (ACL '03)*, 423–430.

Landauer, T. K.; Foltz, P. W.; and Laham, D. 1998. An introduction to latent semantic analysis. *Discourse Processes* 25(2–3):259–284.

Lang, R. R., and Mersch, J. G. 2012. An experiment to determine whether clustering will reveal myemes. In Finlayson, M. A., ed., *Proceedings of the 3rd Workshop on Computational Narrative (CMN'12)*, 20–21.

Lang, R. R. 2010. Considerations in representing mmyth, legends, and folktales. In *Computational Models of Narrative: Papers from the AAAI Fall Symposium*, 29–30.

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

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.

Lin, D. 1998. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998)*, 296–304.

Meadow, C. T.; Kraft, D. H.; and Boyce, B. R. 1999. *Text Information Retrieval Systems*. Orlando, FL: Academic Press, 2nd edition.

Quirk, C.; Brockett, C.; and Dolan, W. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, 142–149.

Resnik, P. 1995. Disambiguating noun groupings with respect to WordNet senses. In Yarowsky, D., and Church, K., eds., *Proceedings of the Third Workshop on Very Large Corpora*, 54–68.

Salton, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Wu, Z., and Palmer, M. 1994. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, 133–138.