Rules for Syntax, Vectors for Semantics

Peter Wiemer-Hastings (Peter.Wiemer-Hastings@ed.ac.uk)
Iraide Zipitria (iraidez@cogsci.ed.ac.uk)
University of Edinburgh
Division of Informatics
2 Buccleuch Place
Edinburgh EH8 9LW Scotland

Abstract

Latent Semantic Analysis (LSA) has been shown to perform many linguistic tasks as well as humans do, and has been put forward as a model of human linguistic competence. But LSA pays no attention to word order, much less sentence structure. Researchers in Natural Language Processing have made significant progress in quickly and accurately deriving the syntactic structure of texts. But there is little agreement on how best to represent meaning, and the representations are brittle and difficult to build. This paper evaluates a model of language understanding that combines information from rule-based syntactic processing with a vector-based semantic representation which is learned from a corpus. The model is evaluated as a cognitive model, and as a potential technique for natural language understanding.

Motivations

Latent Semantic Analysis (LSA) was originally developed for the task of information retrieval, selecting a text which matches a query from a large database (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990)[1]. More recently, LSA has been evaluated by psychologists as a model for human lexical acquisition (Landauer & Dumais, 1997). It has been applied to other textual tasks and found to generally perform at levels matching human performance. All this despite the fact that LSA pays no attention to word order, let alone syntax. This led Landauer to claim that syntax apparently has no contribution to the meaning of a sentence, and may only serve as a working memory crutch for sentence processing, or in a stylistic role (Landauer, Laham, Rehder, & Schreiner, 1997).

The tasks that LSA has been shown to perform well on can be separated into two groups: those that deal with single words and those that deal with longer texts. For example, on the synonym selection part of the TOEFL (Test of English as a Foreign Language), LSA was as accurate at choosing the correct synonym (out of 4 choices) as were successful foreign applicants to US universities (Landauer et al., 1997). For longer texts, Rehder et al (1998) showed that for evaluating author knowledge, LSA does steadily worse for texts shorter than 200 words. More specifically, for 200-word essay segments, LSA accounted for 60% of the variance in human scores. For 60-word essay segments, LSA scores accounted for only 10% of the variance.

In work on judging the quality of single-sentence student answers in an intelligent tutoring context, we have shown in previous work that although LSA nears the performance of intermediate-knowledge human raters, it lags far behind expert performance (Wiemer-Hastings, Wiemer-Hastings, & Graesser, 1999b). Furthermore, when we compared LSA to a keyword-based approach, LSA performed only marginally better (Wiemer-Hastings, Wiemer-Hastings, & Graesser, 1999a). This accords with unpublished results on short answer sentences from Walter Kintsch, personal communication, January 1999.

In the field of Natural Language Processing, the eras of excessive optimism and ensuing disappointment have been followed by study increases in the systems' ability to process the syntactic structure of texts with rule-based mechanisms. The biggest recent developments have been due to the augmentation of the rules with corpus-derived probabilities for when they should be applied (Charniak, 1997; Collins, 1996, 1998, for example).

Unfortunately, progress in the area of computing the semantic content of texts has not been so successful. Two basic variants of semantic theories have been developed. One is based on some form of logic. The other is represented by connections within semantic networks. In fact, the latter can be simply converted into a logic-based representation.

Such theories are brittle in two ways. First, they require every concept and every connection between concepts to be defined by a human knowledge engineer. Multi-purpose representations are not feasible because of the many technical senses of words in every different domain. Second, such representations can not naturally make the graded judgements that humans do. Humans can compare any two things (even apples and oranges!), but aside from counting feature overlap, logic-based representations have difficulty with relationships other than subsumption and "has-as-part".

Due to these various motivations, we are pursuing a two-pronged research project. First, we want to evaluate the combination of a syntactic processing mechanism with an LSA-based semantic representation as a cognitive model of human sentence similarity judgements. Second, we are

[1] We do not describe the functioning of the LSA mechanism here. For a complete description, see (Deerwester et al., 1990; Landauer & Dumais, 1997)

implementing a computational system to automate the processing of texts with this technique. This paper describes the human data we collected for the cognitive modeling aspect, the evaluation of our approach with respect to that data, and our initial attempts to implement the computational system.

Data collection

In (Wiemer-Hastings, 2000), we reported our initial attempts in this direction. In that evaluation, we compared our technique (described more fully below) to human ratings that were previously collected as part of the AutoTutor project (Wiemer-Hastings, Graesser, Harter, & the Tutoring Research Group, 1998). To our surprise, we found that adding syntactic information actually hurt the performance of an LSA-based approach. This could have been due to some problem with the approach, or due to some difficulty with the human data. The previous ratings had been based on complete multi-sentence student answers and ideal good answers. The raters were asked to indicate what percentage of the content of the student answer matched some part of the ideal answer. In the current work, we are looking at similarity ratings for specific sentences. Thus the previous data was not appropriate for our current goals.

To get more relevant human data, we started with text from the AutoTutor Computer Literacy tutoring domain so that we could more directly compare the results with our previous results, and because we had already trained an LSA space for it. AutoTutor “understands” student answers by comparing them to a set of target good answers with LSA. For this evaluation, we randomly paired 300 student answer sentences with 300 target good answers from the relevant questions. We split these into four booklets of 75 pairs, and gave each booklet to four human raters. Because we are also interested in the effect of knowledge on the reliability of ratings, we had previously asked the raters if they were proficient or not with computers. We gave each booklet to two proficient and two non-proficient raters.

We told the raters that the sentence pairs were from the domain of computer literacy, and asked them to indicate how similar the items were on a 6-point scale, from completely dissimilar to completely similar. Here is an example item:

> Sentence 1: ROM only reads information from the disk.
> Sentence 2: The central processing unit, CPU, can read from RAM.

We did not specify how the raters were to decide what similarity means.

The averaged correlations between the human raters were:

> Non-Proficient: $r = 0.35$, $P < 0.001$
> Skilled: $r = 0.45$, $P < 0.001$
> Mean Non-Proficient with Mean Proficient: $r = 0.53$, $P < 0.001$

Although these numbers are relatively low for inter-rater reliability on similarity tasks in general (Tversky, 1977; Goldstone, Medin, & Halberstadt, 1997; Resnik & Diab, 2000, for example), we have found this level of agreement in our other studies of sentence similarity. This task is obviously a difficult one for humans. In future work, we will study the effects of varying the level of context that is available for making these decisions.

Experiment 1: Part-of-speech tags

One way of deriving structural knowledge from text is by performing part of speech tagging. This is one area in which NLP systems have been fairly successful. Brill’s tagger (Brill, 1994) is trained on a marked corpus and uses rules to assign a unique tag to each word. It first assigns the most common tag for each word, then applies a set of automatically-derived context-based rules to modify the original tagging.

When LSA is trained, it does not distinguish between words which are used in multiple parts of speech. This may have significant semantic ramifications. The word “plane”, for example, has very different senses as a verb and as a noun. One way to add structural information to LSA would be to allow it to distinguish the part of speech for each word when training and comparing sentences.

Approach

Our approach to this task was to use the Brill tagger to assign a part-of-speech tag to every word in the training corpus and every word in the test set (which had been given to the human raters). The tag for each word was attached to it with an underscore so that LSA would view each word/tag combination as a single term. For example:

```
ROM_NNP is_VBZ information_NN the_DT
computer_NN was_VBD programmed_VBN with_IN
when_WRB it_PRP was_VBD built_VBN ._.
```

The original training corpus was 2.3 MB of text taken from textbooks and articles on computer literacy. We trained LSA on the tagged corpus at 100, 200, 300, and 400 dimensions because these dimensions had shown reasonable levels of performance in previous evaluations. Then we evaluated this approach by using the new LSA space to calculate the cosines between the tagged versions of the test sentences that had been given to the human raters. We calculated the correlations between the cosines and the human ratings.

Results

Figure 1 shows the correlations between the different LSA models and the human ratings. The first bar depicts the correlation using the standard LSA space (at 200 dimensions) as applied to the untagged versions of the sentences.

Discussion

It is clear that the performance of the tagged models do not match human judgements as closely as the standard

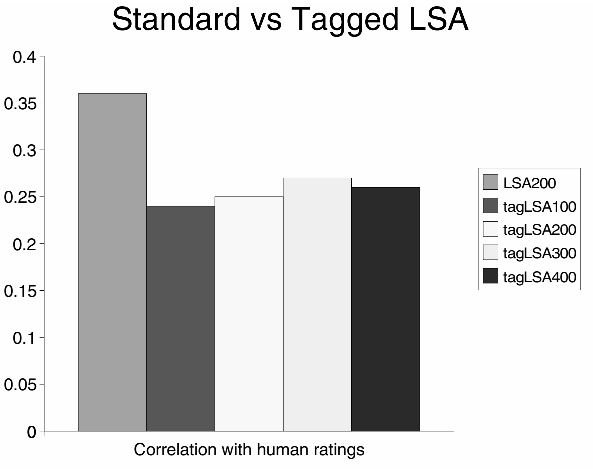

Figure 1: The performance of tagged LSA

LSA approach does. It is not clear why this is. This could support Landauer's claim with respect to the irrelevance of structural information for determining meaning. Perhaps LSA somehow does manage to account for different senses and uses of a word even though it does not have explicit knowledge of the syntactic context of the word's use.

Alternatively, the relatively poor performance could be due to some inadequacies of this particular approach. For example, although the number of dimensions was in the correct relative range with respect to non-tagged LSA processing, perhaps tagged LSA works better with more (or fewer?) dimensions. It could also be that the performance was hampered by mistagging of key words in the sentence. Because the Brill tagger is trained on the Wall Street Journal corpus, its tagging rules often lead it astray when processing the colloquial and domain-specific student answers in our tutoring domain. For example, one student answered a question about a computer's memory like this, "RAM stores things being worked with." The Brill tagger mistagged the word "stores" as a plural noun, thus greatly altering the overall meaning of the sentence.

Experiment 2: Surface parsing

Another obvious potential contribution of sentence structure to meaning is by providing information about the relationships and actions of the participants: the "who did what to whom" information. Although some might claim that LSA is able to derive this information from its training corpus (because men rarely bite dogs, for example), this can not always be the case. And with the exception of case-marked pronouns like "I" and constructions like "there is ...", it is difficult to think of any entity references that can not appear as both subject and object of a sentence. Thus, if we can separately determine the subject, object, and verb parts of a sentence, we should be able to provide information that, in addition to that which we get from LSA, will improve sentence similarity judgements.

The approach: Structured LSA

In standard LSA, the input to the procedure is an entire text, represented as a string. The string is then tokenized into words, and the vector for each word is accessed from the trained vector space. LSA ignores words that it can not find, i.e. those that did not appear in more than one document in the training corpus, or those that appear on the stop-word list, a list of 440 very common words, including most function words. The vector for a text is constructed by simply adding together the relevant word vectors. Two texts are compared by calculating the cosine between their vectors.

In our approach which we call Structured LSA (SLSA), we preprocess input sentences to derive aspects of their structure. More specifically, for each sentence, we:

- resolve pronominal anaphora, replacing pronouns with their antecedents,
- break down complex sentences into simple sentences,
- segment the simple sentences into subject, verb, and object substrings.[2]

.

For example, we transform the student answer: "RAM stores things being worked with, and it is volatile" into:

("stores" "RAM" "things being worked with")
("volatile" "RAM")

This yields a verb string, subject string, and (optional) object string for each sentence. Note that for copular sentences as in the second simple sentence above, the "verb string" is the description following the verb. Also note that our human data was collected not on the original sentences, but on sentences on which the first two steps above were already completed.

SLSA similarity rating

To calculate a similarity score between two sentences with the SLSA approach, the preprocessing is performed on the sentences. Then we separately pass the verb strings, subject strings, and object strings to LSA which computes the cosines between them. Then we average the three together to get an overall similarity rating between the sentences.[3]

Note that this approach provides more information than the standard LSA approach. For each pair of sentences, there are four separate similarity ratings instead of just one.

[2] Passive sentences were normalized, putting the syntactic object as the subject, and vice versa.

[3] In (Wiemer-Hastings, 2000), we evaluated three different methods for aggregating the segment cosines, including a subject-predicate approach and given-new approach. In the current evaluation, the simple average provided the best performance, so we do not present the others here.

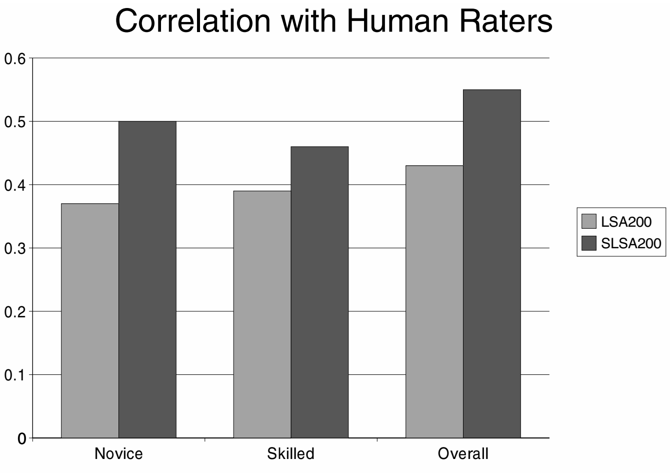

Figure 2: The correlation between SLSA scores and human ratings

In addition to the overall similarity score, SLSA produces separate measures of the similarity of the segments of the sentences. This additional information could be very useful for dialog processing systems.

Results

Because we were interested in evaluating this approach in principle, and not with respect to any particular implementation of the preprocessing technique, we preprocessed the entire test set by hand as described above. Then, the SLSA similarity scores were calculated and correlated with the human ratings. Figure 2 compares the LSA and SLSA approaches with respect to the correlations to human ratings for the non-proficient, proficient, and averaged ratings.

SLSA performed better with respect to each subset of human ratings than did the standard LSA approach. The correlation with the mean of all four human raters was slightly better than the highest level of agreement among the human raters.

Discussion

These results are not consistent with Landauer's claim that syntax does not convey additional semantic content beyond the meanings of individual words. Human sentence similarity judgements are better modelled by an approach that takes structural information into acccount. Although the standard LSA approach does perform as well as humans on longer texts, this may be because the information about who does what to whom in individual sentences is lost in the noise, or is constrained by the larger context.

Toward a hybrid natural language understander

Now that we have validated the benefits of this approach, we have begun to develop a system that will use shallow parsing techniques to automatically perform the preprocessing of input sentences for SLSA. This section describes different approaches that we are evaluating.

Surface parsing

Shallow parsing is currently an area of intense interest in the corpus-based natural language processing community. In fact, the 2001 Computational Natural Language Learning workshop at the Association for Computational Linguistics conference will include a shared task which is to evaluate different techniques for clause splitting. Clause splitting is defined as separating a sentence into subject and predicate parts.

We are currently evaluating the feasibility of using several publicly available surface parsing tools: LTChunk, the SCOL parser, and the Memory-Based Shallow Parser (MBSP). LTChunk was developed by the Language Technology Group at the University of Edinburgh (described at `http://www.ltg.ed.ac.uk/software/chunk/`). It identifies noun phrases and verb groups (combinations of adverbs, auxiliaries, and verbs) in text. The SCOL parser was developed by Abney (1996), and parses text using a set of cascaded rules which delay "difficult" decisions like where to attach prepositional phrases. MBSP (Daelemans, Buchholz, & Veenstra, 1999) is part of the Tilburg Memory Based Learner project (Daelemans, Zavrel, van der Sloot, & van den Bosch, 2000). It is also trained on Penn Treebank Wall Street Journal corpus, performs part-of-speech tagging, and segments texts into subject, verbs, and objects.

Current work

Each of these different methods has drawbacks. The corpus trained approaches have the same difficulty as that noted above: the student answer texts differ sufficiently from the Wall Street Journal to lead to many mistaggings, and therefore, misparses.

Our current efforts are focussing on using the Brill Tagger (adjusting its tags to be more appropriate for our domain), and then the SCOL parser to identify sentence segments. We are developing a postprocessor to transform the output of the parser into the subject, verb, object segmentation that we need as input to SLSA. The postprocessor handles active, passive, and imperative constructions. We are also working on a simple coreference resolution mechanism to allow substitution of antecedents. Our set of hand-processed sentences gives us a useful gold standard against which to evaluate our approach.

The process of matching the segments of the two sentences can be viewed as structure mapping of the type that Gentner et al developed for processing analogies (Gentner, 1983; Forbus, Ferguson, & Gentner, 1994, for example). Ramscar and colleagues have developed a two-stage model for processing analogy which first performs structure-mapping between two scenarios, and then uses LSA to determine the similarity of the slot fillers between the two structures (Yarlett & Ramscar, 2000). For SLSA, the proper treatment of syntactic structures like passives is quite important. Even more difficult are alternations like "give" and "take" which can have the same syntactic structure, but completely different semantic role structures. Resolving such cases seems to require semantic information, resulting in a chicken-and-egg situation. How can we use SLSA to interpret the meaning of a sentence if we must know the meaning in order to use SLSA? Our current research involves treating the verbal and nominal parts of the input sentences differently.

Conclusions

Our findings do not support the claim that syntax provides a negligible contribution to sentence meaning. Instead, a sentence comparison metric that combines structure-derived information with vector-based semantics models human similarity judgements better than LSA alone. As previously mentioned, this approach provides a number of advantages. Its overall fit to human data is not only better than standard LSA, but it provides additional information about the similarity of the different parts of sentences. This could be used in a dialogue-based tutoring system to focus the student's attention on some particular aspect of the target good answer.

With respect to traditional parsing techniques, SLSA has three obvious advantages. First, it is fast, because it does not deal with the combinatorial explosions from ambiguity that most parsers face. Second, it does not require a hand-built semantic concept representation which is tedious to build and brittle. Third, LSA is (in a sense) grounded. Although it does not have direct experience of the world, LSA does have indirect experience via its training corpus. The corpus provides a rich set of interconnections between terms which allows LSA to successfully model many aspects of human linguistic competence.

The limitation of SLSA as a natural language understanding mechanism is that it is only appropriate for tasks where understanding can be cast as computing the similarity of an item to an expected utterance. For tutoring, the approach is feasible because the tutor (whether computer or human) normally determines the topic of conversation, and has some idea of what the student should say. For other tasks where the input utterance is less constrained, this approach might not be the best. On the other hand, if a natural language generation system could be used to generate a set of expected utterances in a particular domain, expectation-based understanding might be feasible and effective.

Although we have presented the syntactic analysis of this work as being derived from symbolic, rule-based mechanisms, our analyses of SLSA as a cognitive model do not depend on this. They would be equally applicable with a connectionist surface parsing technique.

These findings raise quite a few interesting questions for future research. For example:

- What exactly are humans measuring when they rate sentence similarity? Perhaps varying the instructions for human raters will get them to focus on different aspects of meaning.
- What is the best level of granularity to use in segmenting sentences? We have evaluated the use of subject, verb, and object segments, but a coarser or finer segmentation may perform better.
- How much semantic information can be derived from a sentence without knowing its meaning? Inducing additional relationships between the parts of a sentence might improve the SLSA approach, but may require already knowing what the sentence is about.

Addressing these questions will be the focus of our future research.

References

Abney, S. (1996). Partial parsing via finite-state cascades. In *Proceedings of the ESSLLI '96 Robust Parsing Workshop*.

Brill, E. (1994). Some advances in rule-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*. AAAI Press.

Charniak, E. (1997). Statistical Parsing with a Context-free Grammar and Word Statistics. In *Proceedings of the 14th National Conference of the American Association for Artificial Intelligence, Providence, RI., July*, pp. 598–603.

Collins, M. (1996). A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA*, pp. 184–191 San Francisco, CA. Morgan Kaufmann.

Collins, M. (1998). *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Daelemans, W., Buchholz, S., & Veenstra, J. (1999). Memory-Based Shallow Parsing. In *Proceedings of CoNLL-99*.

Daelemans, W., Zavrel, J., van der Sloot, K., & van den Bosch, A. (2000). TiMBL: Tilburg Memory Based Learner, version 3.0, Reference Guide. Tech. rep. Technical Report 00-01, 2000, ILK, University of Tilburg. available at http://ilk.kub.nl/.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

Forbus, K., Ferguson, R., & Gentner, D. (1994). Incremental structure mapping. In *Proceedings of the 16th Annual Conference of the Cognitive Science Society* Mahwah, NJ. Erlbaum.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, *7*, 155–170.

Goldstone, R., Medin, D., & Halberstadt, J. (1997). Similarity in context. *Memory and Cognition*, *25*(2), 237–255.

Landauer, T. K., Laham, D., Rehder, R., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pp. 412–417 Mahwah, NJ. Erlbaum.

Landauer, T., & Dumais, S. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211–240.

Rehder, B., Schreiner, M., Laham, D., Wolfe, M., Landauer, T., & Kintsch, W. (1998). Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes*, *25*, 337–354.

Resnik, P., & Diab, M. (2000). Measuring Verb Similarity. In *Proceedings of the* 22nd *Annual Conference of the Cognitive Science Society* Mahwah, NJ. Erlbaum.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352.

Wiemer-Hastings, P. (2000). Adding syntactic information to LSA. In *Proceedings of the* 22nd *Annual Conference of the Cognitive Science Society*, pp. 989–993 Mahwah, NJ. Erlbaum.

Wiemer-Hastings, P., Graesser, A., Harter, D., & the Tutoring Research Group (1998). The foundations and architecture of AutoTutor. In Goettl, B., Halff, H., Redfield, C., & Shute, V. (Eds.), *Intelligent Tutoring Systems, Proceedings of the 4th International Conference*, pp. 334–343 Berlin. Springer.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. (1999a). How Latent is Latent Semantic Analysis?. In *Proceedings of the Sixteenth International Joint Congress on Artificial Intelligence*, pp. 932–937 San Francisco. Morgan Kaufmann.

Wiemer-Hastings, P., Wiemer-Hastings, K., & Graesser, A. (1999b). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In Lajoie, S., & Vivet, M. (Eds.), *Artificial Intelligence in Education*, pp. 535–542 Amsterdam. IOS Press.

Yarlett, D., & Ramscar, M. (2000). Structure-Mapping Theory and Lexico-Semantic Information. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pp. 571–576 Mahwah, NJ. Erlbaum.