

Combining Independent Modules in Lexical Multiple-Choice Problems

PETER D. TURNEY*, MICHAEL L. LITTMAN**,
JEFFREY BIGHAM*, & VICTOR SHNAYDER**

**NRC, **Rutgers Univ., *Univ. of Washington, **Harvard Univ.*

Abstract

Existing statistical approaches to natural language problems are very coarse approximations to the true complexity of language processing. As such, no single technique will be best for all problem instances. Many researchers are examining ensemble methods that combine the output of multiple modules to create more accurate solutions. This paper examines three merging rules for combining probability distributions: the familiar mixture rule, the logarithmic rule, and a novel product rule. These rules were applied with state-of-the-art results to two problems used to assess human mastery of lexical semantics — synonym questions and analogy questions. All three merging rules result in ensembles that are more accurate than any of their component modules. The differences among the three rules are not statistically significant, but it is suggestive that the popular mixture rule is not the best rule for either of the two problems.

1 Introduction

Asked to articulate the relationship between the words *broad* and *road*, you might consider a number of possibilities. Orthographically, the second can be derived from the first by deleting the initial letter, while semantically, the first can modify the second to indicate above-average width. Many possible relationships would need to be considered, depending on the context. In addition, many different computational approaches could be brought to bear, leaving a designer of a natural language processing system with some difficult choices. A sound software engineering approach is to develop separate modules using independent strategies, then to combine the output of the modules to produce a unified solver.

The concrete problem we consider here is predicting the correct answers to multiple-choice questions. Each instance consists of a context and a finite set of choices, one of which is correct. Modules produce a probability distribution over the choices and a merging rule is used to combine these distributions into one. This distribution, along with relevant utilities, can then be used to select a candidate answer from the set of choices. The merging rules we considered are parameterized, and we set parameters by a maximum likelihood approach on a collection of training instances.

Many problems can be cast in a multiple-choice framework, including optical digit recognition (choices are the 10 digits), word sense disambiguation (choices are a word’s possible senses), text categorization (choices are the classes), and part-of-speech tagging (choices are the grammatical categories). This paper looks at multiple-choice synonym questions (part of the Test of English as a Foreign Language) and multiple-choice verbal analogy questions (part of the SAT college entrance exam). Recent work has shown that algorithms for solving multiple-choice synonym questions can be used to determine the *semantic orientation* of a word; i.e., whether the word conveys praise or criticism (Turney & Littman 2003b). Other research establishes that algorithms for solving multiple-choice verbal analogy questions can be used to ascertain the *semantic relation* in a noun-modifier expression; e.g., in the expression “laser printer”, the modifier “laser” is an *instrument* used by the noun “printer” (Turney & Littman 2003a).

The paper offers two main contributions. First, it introduces and evaluates several new modules for answering multiple-choice synonym questions and verbal analogy questions. Second, it presents a novel product rule for combining module outputs and compares it with other similar merging rules.

Section 2 formalizes the problem addressed in this paper and introduces the three merging rules we study in detail: the mixture rule, the logarithmic rule, and the product rule. Section 3 presents empirical results on synonym problems and Section 4 considers analogy problems. Section 5 summarizes and wraps up.

2 Module combination

The following synonym question is a typical multiple-choice question: hidden:: (a) laughable, (b) veiled, (c) ancient, (d) revealed. The stem, hidden, is the question. There are $k = 4$ choices, and the question writer asserts that exactly one (in this case, (b)) has the same meaning as the stem word. The accuracy of a solver is measured by its fraction of correct answers on a set of ℓ testing instances.

In our setup, knowledge about the multiple-choice task is encapsulated in a set of n modules, each of which can take a question instance and return a probability distribution over the k choices. For a synonym task, one module might be a statistical approach that makes judgments based on analyses of word co-occurrence, while another might use a thesaurus to identify promising candidates. These modules are applied to a training set of m instances, producing probabilistic “forecasts”; $p_{ij}^h \geq 0$ represents the probability assigned by module $1 \leq i \leq n$ to choice $1 \leq j \leq k$ on training instance $1 \leq h \leq m$. The estimated probabilities are distributions of the choices for each module i on each instance h : $\sum_j p_{ij}^h = 1$.

2.1 Merging rules

The rules we considered are parameterized by a set of weights w_i , one for each module. For a given merging rule, a setting of the weight vector w induces a probability distribution over the choices for any instance. Let $D_j^{h,w}$ be the probability assigned by the merging rule to choice j of training instance h when the weights are set to w . Let $1 \leq a(h) \leq k$ be the correct answer for instance h . We set weights to maximize the likelihood of the training data: $w = \operatorname{argmax}_{w'} \prod_h D_{a(h)}^{h,w'}$. The same weights maximize the *mean likelihood*, the geometric mean of the probabilities assigned to correct answers.

We focus on three merging rules in this paper. The *mixture rule* combines module outputs using a weighted sum and can be written $M_j^{h,w} = \sum_i w_i p_{ij}^h$, where $D_j^{h,w} = M_j^{h,w} / \sum_j M_j^{h,w}$ is the probability assigned to choice j of instance h and $0 \leq w_i \leq 1$. The rule can be justified by assuming each instance's answer is generated by a single module chosen via the distribution $w_i / \sum_i w_i$.

The *logarithmic rule* combines the logarithm of module outputs by $L_j^{h,w} = \exp(\sum_i w_i \ln p_{ij}^h) = \prod_i (p_{ij}^h)^{w_i}$, where $D_j^{h,w} = L_j^{h,w} / \sum_j L_j^{h,w}$ is the probability the rule assigns to choice j of instance h . The weight w_i indicates how to scale the module probabilities before they are combined multiplicatively. Note that modules that output zero probabilities must be modified before this rule can be used.

The *product rule* can be written in the form $P_j^{h,w} = \prod_i (w_i p_{ij}^h + (1 - w_i)/k)$, where $D_j^{h,w} = P_j^{h,w} / \sum_j P_j^{h,w}$ is the probability the rule assigns to choice j . The weight $0 \leq w_i \leq 1$ indicates how module i 's output should be mixed with a uniform distribution (or a prior, more generally) before outputs are combined multiplicatively. As with the mixture and logarithmic rules, a module with a weight of zero has no influence on the final assignment of probabilities.

For the experiments reported here, we adopted a straightforward approach to finding the weight vector w that maximizes the likelihood of the data. The weight optimizer reads in the output of the modules, chooses a random starting point for the weights, then hillclimbs using an approximation of the partial derivative. Although more sophisticated optimization algorithms are well known, we found that the simple discrete gradient approach worked well for our application.

2.2 Related work

Merging rules of various sorts have been studied for many years, and have gained prominence recently for natural language applications. Use of the mixture rule and its variations is quite common. Recent examples include the work of Brill & Wu (1998) on part-of-speech tagging, Littman et al. (2002) on crossword-puzzle clues and Florian & Yarowsky (2002) on a word-sense disambiguation task. We

use the name “mixture rule” by analogy to the mixture of experts model (Jacobs et al. 1991), which combined expert opinions in an analogous way. In the forecasting literature, this rule is also known as the linear opinion pool; Jacobs (1995) provides a summary of the theory and applications of the mixture rule in this setting.

The logarithmic opinion pool of Heskes (1998) is the basis for our logarithmic rule. Boosting (Schapire 1999) also uses a logistic-regression-like rule to combine outputs of simple modules to perform state-of-the-art classification. The product of experts approach also combines distributions multiplicatively, and Hinton (1999) argues that this is an improvement over the “vaguer” probability judgments commonly resulting from the mixture rule. A survey by Xu et al. (1992) includes the equal-weights version of the mixture rule. A derivation of the unweighted product rule appears in Xu et al. (1992) and Turney et al. (2003).

An important contribution of the current work is the product rule, which shares the simplicity of the mixture rule and the probabilistic justification of the logarithmic rule. We have not seen an analog of this rule in the forecasting or learning literatures.

3 Synonyms

We constructed a training set of 431 4-choice synonym questions and randomly divided them into 331 training questions and 100 testing questions. We created four modules, described next, and ran each module on the training set. We used the results to set the weights for the three merging rules and evaluated the resulting synonym solver on the test set. Module outputs, where applicable, were normalized to form a probability distribution by scaling them to add to one before merging.

3.1 Modules

LSA. Following Landauer & Dumais (1997), we used latent semantic analysis to recognize synonyms. Our LSA module queried the web interface developed at the University of Colorado (lsa.colorado.edu), which has a 300-dimensional vector representation for each of tens of thousands of words.

PMI-IR. Our Pointwise Mutual Information-Information Retrieval module used the AltaVista search engine (www.altavista.com) to determine the number of web pages that contain the choice and stem in close proximity. PMI-IR used the third scoring method (near each other, but not near not) designed by Turney (2001), since it performed best in this earlier study.

Thesaurus. Our Thesaurus module also used the web to measure word pair similarity. The module queried Wordsmyth (www.wordsmyth.net) and collected any words listed in the “Similar Words”, “Synonyms”, “Crossref. Syn.”,

Synonym solvers	Accuracy	Mean likelihood
LSA only	43.8%	.2669
PMI-IR only	69.0%	.2561
Thesaurus only	69.6%	.5399
Connector only	64.2%	.3757
All: mixture	80.2%	.5439
All: logarithmic	82.0%	.5977
All: product	80.0%	.5889

Table 1: *Comparison of results for merging rules on synonym problems*

and “Related Words” fields. The module created synonym lists for the stem and for each choice, then scored them by their overlap.

Connector. Our Connector module used summary pages from querying Google (`google.com`) with pairs of words to estimate pair similarity. It assigned a score to a pair of words by taking a weighted sum of both the number of times they appear separated by one of the symbols [“,”, “:”, “;”, “=”, “/”, “\”, “(”, “)”, “means”, “defined”, “equals”, “synonym”, “whitespace”, and “and” and the number of times dictionary or thesaurus appear anywhere in the Google summaries.

3.2 Results

Table 1 presents the result of training and testing each of the four modules on synonym problems. The first four lines list the accuracy and mean likelihood obtained using each module individually (using the product rule to set the individual weight). The highest accuracy is that of the Thesaurus module at 69.6%. All three merging rules were able to leverage the combination of the modules to improve performance to roughly 80% — statistically significantly better than the best individual module.

Although the accuracies of the merging rules are nearly identical, the product and logarithmic rules assign higher probabilities to correct answers, as evidenced by the mean likelihood. To illustrate the decision-theoretic implications of this difference, imagine using the probability judgments in a system that receives a score of +1 for each right answer and $-1/2$ for each wrong answer, but can skip questions. In this case, the system should make a guess whenever the highest probability choice is above $1/3$. For the test questions, this translates to scores of 71.0 and 73.0 for the product and logarithmic rules, but only 57.5 for the mixture rule; it skips many more questions because it is insufficiently certain.

3.3 Related work and discussion

Landauer & Dumais (1997) introduced the Test of English as a Foreign Language (TOEFL) synonym task as a way of assessing the accuracy of a learned

Reference	Accuracy	95% confidence
Landauer & Dumais (1997)	64.40%	52.90–74.80%
non-native speakers	64.50%	53.01–74.88%
Turney (2001)	73.75%	62.71–82.96%
Jarmasz & Szpakowicz (2003)	78.75%	68.17–87.11%
Terra & Clarke (2003)	81.25%	70.97–89.11%
Product rule	97.50%	91.26–99.70%

Table 2: *Published TOEFL synonym results*

representation of lexical semantics. Several studies have since used the same data set for direct comparability; Table 2 presents these results.

The accuracy of LSA (Landauer & Dumais 1997) is statistically indistinguishable from that of a population of non-native English speakers on the same questions. PMI-IR (Turney 2001) performed better, but the difference is not statistically significant. Jarmasz & Szpakowicz (2003) give results for a number of relatively sophisticated thesaurus-based methods that looked at path length between words in the heading classifications of Roget’s Thesaurus. Their best scoring method was a statistically significant improvement over the LSA results, but not over those of PMI-IR. Terra & Clarke (2003) studied a variety of corpus-based similarity metrics and measures of context and achieved a statistical tie with PMI-IR and the results from Roget’s Thesaurus.

To compare directly to these results, we removed the 80 TOEFL instances from our collection and used the other 351 instances for training the product rule. The resulting accuracy was statistically significantly better than all previously published results, even though the individual modules performed nearly identically to their published counterparts.

4 Analogies

Synonym questions are unique because of the existence of thesauri — reference books designed precisely to answer queries of this form. The relationships exemplified in analogy questions are quite a bit more varied and are not systematically compiled. For example, the analogy question *cat:meow::* (a) *mouse:scamper*, (b) *bird:peck*, (c) *dog:bark*, (d) *horse:groom*, (e) *lion:scratch* requires that the reader recognize that (c) is the answer because both (c) and the stem are examples of the relation “*X* is the name of the sound made by *Y*”. This type of common sense knowledge is rarely explicitly documented.

In addition to the computational challenge they present, analogical reasoning is recognized as an important component in cognition, including language comprehension (Lakoff & Johnson 1980) and high level perception (Chalmers et al. 1992).

To study module merging for analogy problems, we collected 374 5-choice instances. We randomly split the collection into 274 training instances and 100 testing instances. We next describe the novel modules we developed for attacking analogy problems and present their results.

4.1 Modules

Phrase vectors. We wish to score candidate analogies of the form A:B::C:D (A is to B as C is to D). The quality of a candidate analogy depends on the similarity of the relation R_1 between A and B to the relation R_2 between C and D. The relations R_1 and R_2 are not given to us; the task is to infer these relations automatically. Our approach to this task is to create vectors r_1 and r_2 that represent features of R_1 and R_2 , and then measure the similarity of R_1 and R_2 by the cosine of the angle between the vectors r_1 and r_2 (Turney & Littman 2003a). We create a vector, r , to characterize the relationship between two words, X and Y, by counting the frequencies of 128 different short phrases containing X and Y. Phrases include “X for Y”, “Y with X”, “X in the Y”, and “Y on X”. We use these phrases as queries to AltaVista and record the number of hits (matching web pages) for each query. This process yields a vector of 128 numbers for a pair of words X and Y. The resulting vector r is a kind of *signature* of the relationship between X and Y.

Thesaurus paths. Another way to characterize the semantic relationship, R , between two words, X and Y, is to find a path through a thesaurus or dictionary that connects X to Y or Y to X. In our experiments, we used the WordNet thesaurus (Fellbaum 1998). We view WordNet as a directed graph and the Thesaurus Paths module performed a breadth-first search for paths from X to Y or Y to X. For a given pair of words, X and Y, the module considers all shortest paths in either direction up to three links. It scores the candidate analogy by the maximum degree of similarity between any path for A and B and any path for C and D. The degree of similarity between paths is measured by their number of shared features.

Lexical relation modules. We implemented a set of more specific modules using the WordNet thesaurus. Each module checks if the stem words match a particular relationship in the database. If they do not, the module returns the uniform distribution. Otherwise, it checks each choice pair and eliminates those that do not match. The relations tested are: **Synonym**, **Antonym**, **Hypernym**, **Hyponym**, **Meronym:substance**, **Meronym:part**, **Meronym:member**, **Holonym:substance**, and also **Holonym:member**.

Similarity. Dictionaries are a natural source to use for solving analogies because definitions can express many possible relationships and are likely to make the relationships more explicit than they would be in general text. We employed two definition similarity modules: **Similarity:dict** uses `dictionary.com` and

Analogy solvers	Accuracy	Mean likelihood
Phrase Vectors	38.2%	.2285
Thesaurus Paths	25.0%	.1977
Synonym	20.7%	.1890
Antonym	24.0%	.2142
Hypernym	22.7%	.1956
Hyponym	24.9%	.2030
Meronym:substance	20.0%	.2000
Meronym:part	20.8%	.2000
Meronym:member	20.0%	.2000
Holonym:substance	20.0%	.2000
Holonym:member	20.0%	.2000
Similarity:dict	18.0%	.2000
Similarity:wordsmyth	29.4%	.2058
all: mixture	42.0%	.2370
all: logarithmic	43.0%	.2354
all: product	45.0%	.2512
no PV: mixture	31.0%	.2135
no PV: logarithmic	30.0%	.2063
no PV: product	37.0%	.2207

Table 3: *Comparison of results for merging rules on analogy problems*

Similarity:wordsmyth uses `wordsmyth.net` for definitions. Each module treats a word as a vector formed from the words in its definition. Given a potential analogy $A:B::C:D$, the module computes a vector similarity of the first words (A and C) and adds it to the vector similarity of the second words (B and D).

4.2 Results

We ran the 13 modules described above on our set of training and testing analogy instances, with the results appearing in Table 3 (the product rule was used to set weights for computing individual module mean likelihoods). For the most part, individual module accuracy is near chance level (20%), although this is misleading because most of these modules only return answers for a small subset of instances. Some modules did not answer a single question on the test set. The most accurate individual module was the search-engine-based Phrase Vectors (PV) module. The results of merging all modules was only a slight improvement over PV alone, so we examined the effect of retraining without the PV module. The product rule resulted in a large improvement (though not statistically significant) over the best remaining individual module (37.0% vs. 29.4% for **Similarity:wordsmyth**).

We once again examined the result of deducting 1/2 point for each wrong

answer. The full set of modules scored 31, 33, and 43 using the mixture, logarithmic, and product rules. As in the synonym problems, the logarithmic and product rules assigned probabilities more precisely. In this case, the product rule appears to have a major advantage.

5 Conclusion

We applied three trained merging rules to a set of multiple-choice problems and found all were able to produce state-of-the-art performance on a standardized synonym task by combining four less accurate modules. Although all three rules produced comparable accuracy, the popular mixture rule was consistently weaker than the logarithmic and product rules at assigning high probabilities to correct answers. We provided first results on a challenging verbal analogy task with a set of novel modules that use both lexical databases and statistical information.

In nearly all the tests that we ran, the logarithmic rule and our novel product rule behaved similarly, with a hint of an advantage for the product rule. One point in favor of the logarithmic rule is that it has been better studied so its theoretical properties are better understood. It also is able to “sharpen” probability distributions, which the product rule cannot do without removing the upper bound on weights. On the other hand, the product rule is simpler, executes much more rapidly, and is more robust in the face of modules returning zero probabilities. We feel the strong showing of the product rule proves it worthy of further study.

Acknowledgements. This research was supported in part by a grant from NASA. We’d like to thank the students and TAs of Princeton’s COS302 in 2001 for their pioneering efforts in solving analogy problems; Douglas Corey Campbell, Brandon Braunstein, and Paul Simbi, who provided the first version of our Thesaurus module; Yann LeCun for scanning help, and Haym Hirsh, Philip Resnik, Rob Schapire, Matthew Stone, and Kagan Tumer who served as sounding boards on the topic of module merging.

REFERENCES

- Brill, Eric & Jun Wu. 1998. “Classifier Combination for Improved Lexical Disambiguation”. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL’98)*, vol. 1, 191-195. Montreal, Canada.
- Chalmers, David J., Robert M. French, & Douglas R. Hofstadter. 1992. “High-level Perception, Representation and Analogy: A Critique of Artificial Intelligence Methodology”. *Journal of Experimental and Theoretical Artificial Intelligence* 4:185-211.
- Fellbaum, Christiane. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, Mass.: MIT Press.
- Florian, Radu & David Yarowsky. 2002. “Modeling Consensus: Classifier Combination for Word Sense Disambiguation”. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 25-32. Philadelphia.

- Heskes, Tom. 1998. "Selecting Weighting Factors in Logarithmic Opinion Pools". *Advances in Neural Information Processing Systems* 10:266-272.
- Hinton, Geoffrey E. 1999. "Products of Experts". *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN 99)*, 1:1-6. Edinburgh, Scotland.
- Jacobs, Robert A. 1995. "Methods for Combining Experts' Probability Assessments". *Neural Computation* 7:5.867-888.
- Jacobs, Robert A., Michael I. Jordan, Steve J. Nowlan & Geoffrey E. Hinton. 1991. "Adaptive Mixtures of Experts". *Neural Computation* 3:79-87.
- Jarmasz, Mario & Stan Szpakowicz. 2003. "Roget's Thesaurus and Semantic Similarity". *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, 212-219. Borovets, Bulgaria.
- Lakoff, George & Mark Johnson. 1980. *Metaphors We Live By*. Chicago: University of Chicago Press.
- Landauer, Thomas K. & Susan T. Dumais. 1997. "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge". *Psychological Review* 104:2.211-240.
- Littman, Michael L., Greg A. Keim & Noam Shazeer. 2002. "A Probabilistic Approach to Solving Crossword Puzzles". *Artificial Intelligence* 134:23-55.
- Schapire, Robert E. 1999. "A Brief Introduction to Boosting". *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1401-1406. Stockholm, Sweden.
- Terra, Egidio & C. L. A. Clarke. 2003. "Frequency Estimates for Statistical Word Similarity Measures". *Proceedings of the Human Language Technology and North American Chapter of Association of Computational Linguistics Conference 2003 (HLT/NAACL 2003)*, 244-251. Edmonton, Alberta, Canada.
- Turney, Peter D. 2001. "Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL". *Proceedings of the 12th European Conference on Machine Learning (ECML-2001)*, 491-502. Freiburg, Germany.
- Turney, Peter D. & Michael L. Littman. 2003a. "Learning Analogies and Semantic Relations". Technical Report ERB-1103. Ottawa, Ontario, Canada: National Research Council, Institute for Information Technology.
- Turney, Peter D. & Michael L. Littman. 2003b. "Measuring Praise and Criticism: Inference of Semantic Orientation from Association". *ACM Transactions on Information Systems* 21:4.315-346.
- Turney, Peter D., Michael L. Littman, Jeffrey Bigham & Victor Shnayder. 2003. "Combining Independent Modules to Solve Multiple-Choice Synonym and Analogy Problems". *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*, 482-489. Borovets, Bulgaria.
- Xu, Lei, Adam Krzyzak & Ching Y. Suen. 1992. "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition". *IEEE Transactions on Systems, Man and Cybernetics* 22:3.418-435.