Taking the SAT Exam: Sentence Completion

Katherine Busch
Stanford University
kbusch@stanford

Caitlin Colgrove
Stanford University
colgrove@stanford

Julia Neidert
Stanford University
jneid@stanford

Abstract

In this project we implement a system for automatically answering SAT sentence completion (“cloze”) questions. We experiment with three different approaches, which make use of trigram language models, positive pointwise mutual information, and WordNet glosses and relations. Each of these methods shows a significant improvement over the random baseline. However, simple attempts to combine methods show little to no improvement over the best of the individual methods. Even so, our best systems demonstrate performance comparable to the average human test-taker. Additionally, we saw a significant difference in performance between each of the models on our development set (taken from test-prep websites) and on the test set (from official SAT practice tests).

1 Introduction

Every year, countless high school students take the SAT Reasoning Test , an examination meant to assess the students preparedness for college. The Critical Reading section of the exam serves as a test of natural language understanding, and can thus be used to evaluate an automatic NLU system. One specific type of problem on the exam is the Sentence Completion problem. This problem is a version of a cloze test, in which the participant must replace missing words in a text. On the SAT, the sentence completion task is, given a sentence from which one or two words have been replaced by a blank, to fill in the blank or blanks by choosing one of five alternate words or word pairs. Consider the following example, taken directly from the SAT website (sat.collegeboard.org):

```
Because King Philip’s desire to make
Spain the dominant power in sixteenth-
century Europe ran counter to Queen
Elizabeth’s insistence on autonomy
for England, ------- was -------.

(A) reconciliation..assured
(B) warfare..avoidable
(C) ruination..impossible
(D) conflict..inevitable
(E) diplomacy..simple
```

We worked to build an automatic system for achieving this sentence completion task.

To our knowledge, no one has addressed this exact problem before. However, notable previous approaches to fill-in-the-blank tasks (typically with near-synonyms) have included lexical co-occurrence networks (Edmonds, 1997), statistical models using PMI (Inkpen, 2007), n-gram language models (Islam and Inkpen, 2010), and vector space modelling (Wang and Hirst, 2010). The approach of lexical co-occurrence network creates a graph with edges between words weighted by the t-score between the two words and scores sentences by the sum of the weights of connected words. The statistical approach computes the PMI of the word in the blank with the words in the rest of the sentence and takes the solution with the highest sum. The N-gram language model approach is similar, but it computes the most likely sentences based on the probability of 5-grams. Finally, the vector space model creates a word by word matrix over the corpus, then represents the sentence as a binary vector and chooses the word with the highest cosine similarity.

There have also been some attempts at other SAT-type questions such as (Turney and Littman, 2003) and (Turney, 2006). These papers look at the SAT analogy questions and try to model the relationships between words. The best score, from (Turney, 2006), is 55.7%, comparable to the average student score.

We approached the problem from three main angles, creating a trigram language model, a PPMI model, and a model that uses WordNet relations. Finally, we attempted to integrate these three models into a superior combined one on the hypothesis that each specialized at answering a different subset of questions.

2 Data

2.1 SAT Questions

In order to test our implementation, we gathered 408 practice SAT sentence completion questions from 12 different online test-preparation sites, including collegeboard.org, princetonreview.com, and kaplan.com (see References for a complete list). The quality and similarity to actual SAT questions varies among these sources, so for our final test set we used 190 actual practice SAT questions from the 2009 Official SAT Study Guide published by the College Board, the creator of the exam.

2.2 Additional Data

To train a language model and extract mutual information data, we used the New York Times section of the Gigaword Corpus (Graff, 2003), which includes all English New York Times articles from July 1994 through December 2006. We stripped the text of XML using simple regular expressions, and split it into sentences and tokens using the Stanford Core NLP sentence splitter.

We also used the WordNet (Miller, 1995) dictionary for term glosses, synsets, and relations.

3 Methodology

To solve the sentence completion problem, we focused on three distinct approaches. The first looks at the likelihood of each possible completion given a language model trained on the Gigaword New York Times corpus. The second looks at the positive pointwise mutual information between the answer choices and the rest of the sentence. Finally, we looked at overlap of the senntence words with glosses and relations in WordNet for the various answer choices. All of these approaches were compared to a simple random baseline.

3.1 Language Model

We constructed a language model using the SRI Language Modeling Toolkit (Stolcke, 2002) and used this model to score the probability of a sentence. We created the language model using both 5-grams and trigrams from the first 30 New York Times files of the Gigaword corpus with Good-Turing discounting and Katz backoff for smoothing. We then plugged each of the five candidate answer choices into the sentence and calculated the probability of that sentence given the language model. We chose the most likely sentence from this model as the answer.

3.2 PPMI

The positive pointwise mutual information between two words is calculated by the following formula:

$$\text{PPMI}(w1, w2) = \max(\log \frac{P(w1, w2)}{P(w1)P(w2)}, 0)$$

where $P(w)$ is the probability of a word w appearing in a particular sentence, and $P(w1, w2)$ is the probability of two words, $w1$ and $w2$, appearing in the same sentence. These probabilities were calculated by counting the occurrences of words over 50 months of New York Times articles. To reduce the space needed for this operation, we only kept counts for words that appeared in either a question or a possible answer.

Because some SAT words are unusual, we faced some sparsity challenges with regards to word counts. If $P(w1, w2)$ was zero, but both of the probabilities $P(w1)$ and $P(w2)$ were non-zero, the PMI would be negative infinity, so we simply assigned this case a score of 0. Worse, because the SAT tends to use unusual words to test vocabulary, sometimes the individual probabilities $P(w)$ were also zero. This would lead to an undefined PMI score, which does not have a defined interpretation in the formula given above. However, as extremely rare words should not contribute much to the likelihood of a sentence, we decided to also assign these cases a score of zero. We did not otherwise smooth the scores.

After generating the PPMI for every word pair, we assigned a score to each of the five possible completions according to the following metric, which is similar to Inkpen (2007). For a given completed sentence S, let S_A be the set of words appearing in the answer, and S_Q be the set of words appearing in the sentence. Then

$$Score_{\text{PPMI}}(S) = \sum_{w1 \in S_A} \sum_{w2 \in S_Q} \text{PPMI}(w1, w2)$$

Because all possible answers have the same number of blanks and the same number of words per blank, the outer sum is simply over the set of all words appearing in each answer.

Finally, for each question we selected the answer with the highest PPMI score for the completed sentence.

3.3 Dictionary

We noticed that many of the SAT sample questions contained words directly related to the definitions of the words in the blanks. To address this, we considered the glosses of the answer choices, as given in the WordNet (Miller, 1995) dictionary, and computed the overlap of these words with the words in the question sentence. To provide additional information, for each answer term, we also considered the words in its synset, as well as all of the words related to it in WordNet by any direct relation. To decrease sparsity, we used the Porter Stemmer (Van Rijsbergen, 1980) to match any two terms with the same stem. The Natural Language Toolkit (Bird, 2009) was used to interface with WordNet and the Porter Stemmer. The best answer was then the answer choice with the maximum overlap of its gloss, synset, and relation terms with the question sentence words.

3.4 Negation

We identified when a word should be considered negated using a technique inspired by (Das and Chen, 2001). For each word in the set [“never, “no”, “nothing”, “nowhere”, “noone”, “none”, “not”, “n’t”, “but”, “however”, “yet”], we marked each word following it as negative until we arrived at punctuation in the set [“.”, “,”, “:”, “;”, “!”, “?”]. When we arrived at a second negation word before a punctuation mark, we flipped the negation flag, no longer marking the words as negative until we encountered another word in the above set.

3.5 Combined Models

Because initial analysis of results showed that different models accurately answered significantly different sets of questions, we experimented with several methods of combining the results to improve our system.

3.5.1 Simple Classifier

Our first strategy was to take the the three scores generated for each completed sentence and apply a simple logistic regression classifier to the values. We defined our classification problem as classifying a possible completion as “correct” or “incorrect.” Then, to choose a single correct answer, we chose the answer with the highest score for the “correct” class. For features, we tried both normalized values and unnormalized values of the scores, as well as experimenting with a balanced data set (equal numbers of correct and incorrect sentence choices).

3.5.2 Sum of Confidence Scores

Second, we attempted to generate “confidence scores for each question’s possible completions. These scores were typically taken by normalizing the probabilities over the possible answer choices, to give each answer choice a score between zero and one. To answer a question, we took the answer with the highest sum of confidence scores. We also experimented with taking the sentence with the highest maximum confidence over the three scores.

3.5.3 Weighted Sum of Confidence Scores

Finally, we used a weighted linear combination of the scores to select the best answer choice. Since finding the correct answer to a sentence completion question requires taking a maximum over several values, we could not use an optimization algorithm such as gradient ascent to optimize the weights for each score. Instead, we found the weights using a grid search. Beginning with a log-separated set of possible values for each weight (0, 0.0001, 0.001, 0.01, 0.1, 1), we evaluated the accuracy for every possible combination of these weights and found the combination of weights that maximized accuracy.

Model	Accuracy (Development)	Accuracy (Test)	Significance (random baseline) [1]
Language Model	40.0%	32.1%	0.00799
PPMI	51.0%	46.3%	0.000999
Dictionary	29.9%	42.6%	0.000999

Table 1: Performance for individual models.

Model	Accuracy (Development)	Accuracy (Test)	Significance (compared to PMI) [1]
Oracle	77.0%	76.3%	—
Voting	48.2%	45.3%	0.869
Max Sum Confidence	54.2%	48.9%	0.373
Weighted Sum (weights from dev set)	54.9%	45.8%	1.000
Weighted Sum (optimal)	—	54.7%	0.0270

Table 2: Performance for combined models.

For each score type, we then selected a new set of potential weights near the optimum weight found before, and maximized accuracy with these new sets of possible weights. This process was repeated until all weights were found to two significant digits. To evaluate this procedure, we ran 10-fold cross-validation, learning weights on 90% of the questions and evaluating the accuracy on the remaining 10% for each of the 10 partitions of the SAT questions.

3.6 Test Taking Strategy

On the SAT, correct answers earn 1 point, incorrect answers earn -0.25 points, and skipped answers earn 0 points. We considered skipping questions on which we had low confidence to improve our SAT score. However, since all of our models answered correctly with probability more than 25%, any guess had a positive expected point value. Therefore, the optimal score was achieved by guessing on every question.

4 Results

4.1 Individual Approaches

Each of our individual approaches showed a statistically significant improvement in performance over a random baseline (which will be correct about 20% of the time), as can be seen in Table 1. On both the development and the test set, the PPMI approach had the highest accuracy, though it performed slightly worse on the test set. On the development set, the 5-gram and trigram language models both performed similarly, so we chose to use the trigram model to decrease running time. The language model also performed substantially worse on the test data than on the development set. Even more surprising, however, was the huge increase in performance of the dictionary method in the test set compared to the development set.

One possible hypothesis for the differing results between data sets is that the official SAT questions are often designed to test relatively obscure vocabulary. As a result, the most statistically likely word from the statistical approaches (language model and PPMI) may actually be less likely to be a vocabulary word and thus less likely to be the correct answer. This problem is exacerbated for the trigram language model because so little context is considered. PPMI, on the other hand, will do a better job at capturing the overall meaning of the sentence and thus have a better chance at finding the appropriate word, even if that word is unlikely.

Moreover, the large improvement in the dictionary score is also likely due to the differing data sets.

[1] We performed statistical significance testing using approximate randomization with R=1000.

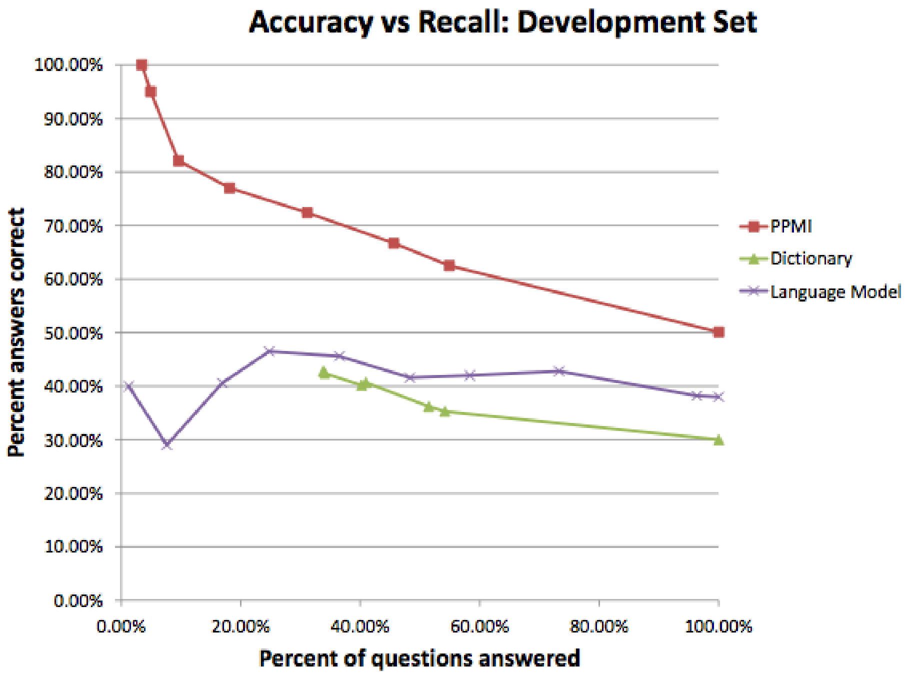

Figure 1: Performance of each model restricted to questions for which the model's confidence was above a certain threshold (on the development set)

.

When we manually examined the data, we found that a large number of the official SAT questions had a similar form, in which the vocabulary word to be tested was in the answer choices, and some form of the definition of this word was contained in the sentence itself. Because of the large number of such questions, the dictionary method (which we designed to address this particular type of question) does quite well.

Additionally, we experimented on sets of questions with just one blank or just two blanks to see if different models were better at different types of questions. In the development set, PPMI performed marginally better on one blank questions (52.7% versus 48.5% accuracy) and the language model performed marginally better on two blank questions (40% versus 36.6% accuracy). However, we did see a larger difference for the dictionary scores: the dictionary scores achieved 30.5% accuracy on one blank questions compared to only 24.8% accuracy on two blank questions in our development set. One possible explanation for this difference is that questions that are asking about definitions of a particular vocabulary word will likely have only one blank.

4.2 Confidence Measures

In trying to combine scores, we experimented with using a confidence metric for the scores from each model. Ideally, if we had an accurate confidence metric, we could use this to choose which models scores to use, hopefully getting us closer to the upper bound given by the oracle. On the development set (Figure 1), our confidence score for PPMI is nicely correlated with the accuracy of the model tested only questions for which the confidence score was above a minimum threshold. Unfortunately, on this set, our dictionary and language model scores did not show as much of a correlation, barely changing when the confidence threshold was increased. However, on the test set of official questions (Figure 2), the confidence scores of the dictionary and language models have a much better relationship with the accuracy of the predictions, indicating cleaner data.

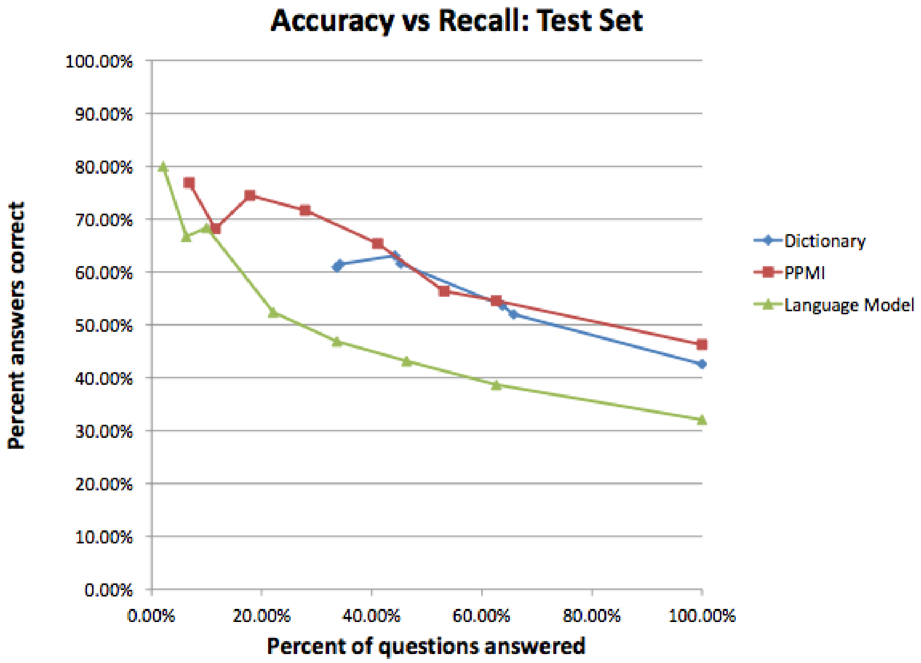

Figure 2: Performance of each model restricted to questions for which the model's confidence was above a certain threshold (on the test set)

.

4.3 Combined Approaches

Each of our approaches individually boosts performance, and each seems to succeed at answering different questions. We found that theoretically, if we were able to use an oracle to choose the correct answer given that at least one of our three approaches had selected it, we would achieve an accuracy of 76.3%. However, combining the approaches effectively to achieve this proved difficult.

Using a classifier to select the correct answer turned out to be ineffective, resulting in essentially random performance. This is perhaps because accuracy in this case is not a continuous function since it involves taking a max, making it difficult to fit parameters to it.

A simple voting mechanism, where the choice most frequently selected among the three approaches was chosen, actually resulted in worse performance than using PPMI scores alone. Choosing the answer with the maximum sum of confidence scores increased accuracy a bit over PPMI, but not significantly.

Using grid search to find weights for a linear combination of each approachs confidence score gave mixed results. Computing these weights on the development set and applying them to the test set resulted in worse performance than a simple sum. However, training weights on the test set significantly improved results from PPMI's 46.3% to 54.7%. As discussed above, this was due to the differing levels of performance of each approach in the development set and the test set. Unsurprisingly, the development set weights were 0.47 for the PPMI score, 0.5 for the language model score, and 0.03 for the dictionary score, while the optimal test set weights were only 0.21 for PPMI, 0.37 for the language model, and 0.42 for the dictionary scores. Note that the PPMI weighting is actually lower than the language model weight for the test set even though PPMI performed better individually, likely due to the greater absolute confidence that PPMI scores provide. This discrepancy between the development and test results shows both that we were overfitting these parameters on our development set, and that the development set was not a very good representation of actual SAT questions.

Since we observed differing performance of the various approaches between one-blank and two-blank questions, we also trained separate weights for each of these types of questions. This was able to give the dictionary scores greater weight for one-blank questions than for two-blank questions, especially in the development weights, but did not significantly change the accuracy of the system.

5 Discussion and Conclusion

Overall, we were able to improve from random guessing to near average human performance. Supposing that all questions on the SAT Critical Reading section were sentence completion questions, we would score 470 out of 800 using our maximum sum confidence system, which is in the 40th percentile of student performance (College Entrance Examination Board, 2011). Each of our approaches, based on language models, PPMI, and dictionary features, contributed to this result, although PPMI is especially useful.

Our main concern is the poor quality of example SAT questions available on the Internet. Developing a system on example questions from 12 different sites resulted in poor performance on a test set of questions from actual SAT practice questions. If our development data had been representative of the actual test data, using the weighted sum method would have resulted in an increase in accuracy of more than 5%, the equivalent of scoring 490 on the SAT Critical Reading section (47th percentile). This suggests that the types of resources used to prepare for tests like the SAT can actually make a significant difference in performance.

In addition to using better data for preparation, this work could be improved in several areas. One significant consideration is the relationship between the blanks in a two-blank question. Features like discourse relations, sentiment, and antonymy could be used to model this relationship. Since our system generally performs worse on two-blank questions than one-blank questions, this improvement could make a significant difference.

We could also try to improve the combination of features by finding confidence scores that more accurately represent the likelihood of correctly answering a question. One approach to this is to fit a line to the relationship between raw scores and the probability of getting a question correct, using this mapping to get an empirical confidence weighting. A final improvement could be gained from attempting to identify the words or types of words in the sentence most most relevant to the query. Using parse trees to find words directly related to the missing terms might provide more precise information.

While getting a perfect score on the SAT would be an admirable achievement, this system can be applied to more useful tasks. Any application that involves noisy or error-prone data, such as optical character recognition, automatic essay grading, and grammar correction could make use of a system for finding the best word to use at a particular spot in a sentence. Similarly, this can be applied to creating sentences in natural language generation or machine translation, or to evaluating these sentences. In this way, a sentence completion system would not only be prepared for college, but also be ready to use language in the natural world.

References

Bird, Steven, Edward Loper and Ewan Klein 2009. *Natural Language Processing with Python.* O'Reilly Media Inc.

College Entrance Examination Board 2011. *SAT Percentile Ranks.* http://media.collegeboard.com/digitalServices/pdf/SAT-Percentile_Ranks_2011.pdf.

Edmonds, Philip. 1997. Choosing the Word Most Typical in Context Using a Lexical Co-occurrence Network. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 507509, Madrid, Spain. Association for Computational Linguistics. http://www.aclweb.org/anthology/P97-1067.

Inkpen, Diana 2007. A statistical model for near-synonym choice. *ACM Trans. Speech Lang. Process.*, Process. 4, 1, Article 2 (February 2007), 17 pages. Association for Computational Linguistics. http://doi.acm.org/10.1145/1187415.1187417.

Islam, Aminul and Diana Inkpen 2010. Near-Synonym Choice using a 5-gram Language Model. *Research in Computing Science: Special issue on Natural Language Processing and its Applications*, 46, 41-52. Association for Computational Linguistics. http://www.site.uottawa.ca/~mdislam/publications/RCS_2010.pdf.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, Vol. 38, No. 11: 39-41.

Sanjiv Das and Mike Chen 2001. Yahoo! for Amazon: extracting market sentiment from stock message boards. *In Proceedings of the 8th Asia Pacific Finance Association Annual Conference*.

The Stanford NLP Group 2012. *Stanford CoreNLP version 1.3.1*.

Stolcke, Andreas. SRLIM – An Extensible Language Model Toolkit. *ICSLP-2002, 901-904*.

Turney, Peter and Michael Littman 2003. Learning Analogies and Semantic Relations. *National Research Council of Canada*.

Turney, Peter 2006. Expressing Implicit Semantic Relations without Supervision. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 313320, Sydney.

Van Rijsbergen, C.J., S.E. Robertson and M.F. Porter 1980. New models in probabilistic information retrieval. *British Library Research and Development Report, no. 5587*, London: British Library.

Wang, Tong and Graeme Hirst 2010. Near-synonym Lexical Choice in Latent Semantic Space. *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 11821190, Beijing, China. Coling 2010 Organizing Committee. `http://www.aclweb.org/anthology/C10-1133`.

Appendix: SAT Question Sources

- `http://sat.jumbotests.com/tests/sat-sentence-completion-questions`
- `http://www.act-sat-prep.com/completionx.html`
- `http://sat.collegeboard.org/practice/sat-practice-questions-reading/sentence-completion`
- `http://www.proprofs.com/sat/practice-questions.shtml`
- `http://collegeapps.about.com/library/sentence_completion_1/bl_sentcomp1_quiz.htm`
- `https://satonlinecourse.collegeboard.com/SR/digital_assets/assessment/pdf/F4D31AB0-66B4-CE32-00F7-F5405701F413-F.pdf`
- `http://www.majortests.com/sat/sentence-completion.php`
- `http://www.snapwiz.com/prep/sat/sentence-completion.html`
- `http://www.sattest.us/verbal/sentence_completion/sentence_completion_questions.php`
- `http://www.testprepauthority.com/SAT-blog/bid/48280/SAT-Critical-Reading-Practice-//Question-1-Sentence-Completion`
- `http://learning.princetonreview.com/courses/SAT/SAT_Online_Test_3/crs-client.htmhttp://216.154.212.161/KaplanQuizzes/showQuiz.jsp?TID=SATCRPT1`
- College Entrance Examination Board. (2009). The official SAT study guide. New York: College Board .