Computational Approaches to Sentence Completion

Geoffrey Zweig, John C. Platt
Christopher Meek
Christopher J.C. Burges
Microsoft Research
Redmond, WA 98052

Ainur Yessenalina
Cornell University
Computer Science Dept.
Ithaca, NY 14853

Qiang Liu
Univ. of California, Irvine
Info. & Comp. Sci.
Irvine, California 92697

Abstract

This paper studies the problem of sentence-level semantic coherence by answering SAT-style sentence completion questions. These questions test the ability of algorithms to distinguish sense from nonsense based on a variety of sentence-level phenomena. We tackle the problem with two approaches: methods that use local lexical information, such as the n-grams of a classical language model; and methods that evaluate global coherence, such as latent semantic analysis. We evaluate these methods on a suite of practice SAT questions, and on a recently released sentence completion task based on data taken from five Conan Doyle novels. We find that by fusing local and global information, we can exceed 50% on this task (chance baseline is 20%), and we suggest some avenues for further research.

1 Introduction

In recent years, standardized examinations have proved a fertile source of evaluation data for language processing tasks. They are valuable for many reasons: they represent facets of language understanding recognized as important by educational experts; they are organized in various formats designed to evaluate specific capabilities; they are yardsticks by which society measures educational progress; and they affect a large number of people.

Previous researchers have taken advantage of this material to test both narrow and general language processing capabilities. Among the narrower tasks, the identification of synonyms and antonyms has been studied by (Landauer and Dumais, 1997; Mohammed et al., 2008; Mohammed et al., 2011; Turney et al., 2003; Turney, 2008), who used questions from the Test of English as a Foreign Language (TOEFL), Graduate Record Exams (GRE) and English as a Second Language (ESL) exams. Tasks requiring broader competencies include logic puzzles and reading comprehension. Logic puzzles drawn from the Law School Administration Test (LSAT) and the GRE were studied in (Lev et al., 2004), which combined an extensive array of techniques to solve the problems. The DeepRead system (Hirschman et al., 1999) initiated a long line of research into reading comprehension based on test prep material (Charniak et al., 2000; Riloff and Thelen, 2000; Wang et al., 2000; Ng et al., 2000).

In this paper, we study a new class of problems intermediate in difficulty between the extremes of synonym detection and general question answering - the sentence completion questions found on the Scholastic Aptitude Test (SAT). These questions present a sentence with one or two blanks that need to be filled in. Five possible words (or short phrases) are given as options for each blank. All possible answers except one result in a nonsense sentence. Two examples are shown in Figure 1.

The questions are highly constrained in the sense that all the information necessary is present in the sentence itself without any other context. Nevertheless, they vary widely in difficulty. The first of these examples is relatively simple: the second half of the sentence is a clear description of the type of behavior characterized by the desired adjective. The second example is more sophisticated; one must infer from

1. One of the characters in Milton Murayama's novel is considered _____ because he deliberately defies an oppressive hierarchical society.
(A) rebellious (B) impulsive (C) artistic (D) industrious (E) tyrannical

2. Whether substances are medicines or poisons often depends on dosage, for substances that are _____ in small doses can be _____ in large.
(A) useless .. effective
(B) mild .. benign
(C) curative .. toxic
(D) harmful .. fatal
(E) beneficial .. miraculous

Figure 1: Sample sentence completion questions (Educational-Testing-Service, 2011).

the contrast between medicine and poison that the correct answer involves a contrast, either *useless vs. effective* or *curative vs. toxic*. Moreover, the first, incorrect, possibility is perfectly acceptable in the context of the second clause alone; only irrelevance to the contrast between medicine and poison eliminates it. In general, the questions require a combination of semantic and world knowledge as well as occasional logical reasoning. We study the sentence completion task because we believe it is complex enough to pose a significant challenge, yet structured enough that progress may be possible.

As a first step, we have approached the problem from two points-of-view: first by exploiting local sentence structure, and secondly by measuring a novel form of global sentence coherence based on latent semantic analysis. To investigate the usefulness of local information, we evaluated n-gram language model scores, from both a conventional model with Good-Turing smoothing, and with a recently proposed maximum-entropy class-based n-gram model (Chen, 2009a; Chen, 2009b). Also in the language modeling vein, but with potentially global context, we evaluate the use of a recurrent neural network language model. In all the language modeling approaches, a model is used to compute a sentence probability with each of the potential completions. To measure global coherence, we propose a novel method based on latent semantic analysis (LSA). We find that the LSA based method performs best, and that both local and global information can be combined to exceed 50% accuracy. We report results on a set of questions taken from a collection of SAT practice exams (Princeton-Review, 2010), and further validate the methods with the recently proposed MSR Sentence Completion Challenge set (Zweig and Burges, 2011).

Our paper thus makes the following contributions: First, we present the first published results on the SAT sentence completion task. Secondly, we evaluate the effectiveness of both local n-gram information, and global coherence in the form of a novel LSA-based metric. Finally, we illustrate that the local and global information can be effectively fused.

The remainder of this paper is organized as follows. In Section 2 we discuss related work. Section 3 describes the language modeling methods we have evaluated. Section 4 outlines the LSA-based methods. Section 5 presents our experimental results. We conclude with a discussion in Section 6.

2 Related Work

The past work which is most similar to ours is derived from the lexical substitution track of SemEval-2007 (McCarthy and Navigli, 2007). In this task, the challenge is to find a replacement for a word or phrase removed from a sentence. In contrast to our SAT-inspired task, the original answer is indicated. For example, one might be asked to find alternates for *match* in "After the *match*, replace any remaining fluid deficit to prevent problems of chronic dehydration throughout the tournament." Two consistently high-performing systems for this task are the KU (Yuret, 2007) and UNT (Hassan et al., 2007) systems. These operate in two phases: first they find a set of potential replacement words, and then they rank them. The KU system uses just an N-gram language model to do this ranking. The UNT system uses a large variety of information sources, and a language model score receives the highest weight. N-gram statistics were also very effective in (Giuliano et al., 2007). That paper also explores the use of Latent Semantic Analysis to measure the degree of similarity between a potential replacement and its context, but the results are poorer than others. Since the original word provides a strong hint as to the pos-

sible meanings of the replacements, we hypothesize that N-gram statistics are largely able to resolve the remaining ambiguities. The SAT sentence completion sentences do not have this property and thus are more challenging.

Related to, but predating the Semeval lexical substitution task are the ESL synonym questions proposed by Turney (2001), and subsequently considered by numerous research groups including Terra and Clarke (2003) and Pado and Lapata (2007). These questions are similar to the SemEval task, but in addition to the original word and the sentence context, the list of options is provided. Jarmasz and Szpakowicz (2003) used a sophisticated thesaurus-based method and achieved state-of-the art performance, which is 82%.

Other work on standardized tests includes the synonym and antonym tasks mentioned in Section 1, and more recent work on a SAT analogy task introduced by (Turney et al., 2003) and extensively used by other researchers (Veale, 2004; Turney and Littman, 2005; D. et al., 2009).

3 Sentence Completion via Language Modeling

Perhaps the most straightforward approach to solving the sentence completion task is to form the complete sentence with each option in turn, and to evaluate its likelihood under a language model. As discussed in Section 2, this was found be be very effective in the ranking phase of several SemEval systems. In this section, we describe the suite of state-of-the-art language modeling techniques for which we will present results. We begin with n-gram models; first a classical n-gram backoff model (Chen and Goodman, 1999), and then a recently proposed class-based maximum-entropy n-gram model (Chen, 2009a; Chen, 2009b). N-gram models have the obvious disadvantage of using a very limited context in predicting word probabilities. Therefore we evaluate the recurrent neural net model of (Mikolov et al., 2010; Mikolov et al., 2011b). This model has produced record-breaking perplexity results in several tasks (Mikolov et al., 2011a), and has the potential to encode sentence-span information in the network hidden-layer activations. We have also evaluated the use of parse scores, using an off-the-shelf stochastic context free grammar parser. However, the grammatical structure of the alternatives is often identical. With scores differing only in the final non-terminal/terminal rewrites, this did little better than chance. The use of other syntactically derived features, for example based on a dependency parse, are likely to be more effective, but we leave this for future work.

3.1 Backoff N-gram Language Model

Our baseline model is a Good-Turing smoothed model trained with the CMU language modeling toolkit (Clarkson and Rosenfeld, 1997). For the SAT task, we used a trigram language model trained on 1.1B words of newspaper data, described in Section 5.1. All bigrams occurring at least twice were retained in the model, along with all trigrams occurring at least three times. The vocabulary consisted of all words occurring at least 100 times in the data, along with every word in the development or test sets. This resulted in a 124k word vocabulary and 59M n-grams. For the Conan Doyle data, which we henceforth refer to as the *Holmes data* (see Section 5.1), the smaller amount of training data allowed us to use 4-grams and a vocabulary cutoff of 3. This resulted in 26M n-grams and a 126k word vocabulary.

3.2 Maximum Entropy Class-Based N-gram Language Model

Word-class information provides a level of abstraction which is not available in a word-level language model; therefore we evaluated a state-of-the-art class based language model. Model M (Chen, 2009a; Chen, 2009b) is a recently proposed class based exponential n-gram language model which has shown improvements across a variety of tasks (Chen, 2009b; Chen et al., 2009; Emami et al., 2010). The key ideas are the modeling of word n-gram probabilities with a maximum entropy model, and the use of word-class information in the definition of the features. In particular, each word w is assigned deterministically to a class c, allowing the n-gram probabilities to be estimated as the product of class and word parts

$$P(w_i|w_{i-n+1}\dots w_{i-2}w_{i-1}) = \\ P(c_i|c_{i-n+1}\dots c_{i-2}c_{i-1}, w_{i-n+1}\dots w_{i-2}w_{i-1}) \\ P(w_i|w_{i-n+1}\dots w_{i-2}w_{i-1}, c_i).$$

Both components are themselves maximum entropy n-gram models in which the probability of a word or class label l given history h is determined by $\frac{1}{Z}\exp(\sum_k f_k(h,l))$. The features $f_k(h,l)$ used are the presence of various patterns in the concatenation of hl, for example whether a particular suffix is present in hl.

3.3 Recurrent Neural Net Language Model

Many of the questions involve long-range dependencies between words. While n-gram models have no ability to explicitly maintain long-span context, the recently proposed recurrent neural-net model of (Mikolov et al., 2010) does. Related approaches have been proposed by (Sutskever et al., 2011; Socher et al., 2011). In this model, a set of neural net activations $\mathbf{s}(t)$ is maintained and updated at each sentence position t. These activations encapsulate the sentence history up to the t^{th} word in a real-valued vector which typically has several hundred dimensions. The word at position t is represented as a binary vector $\mathbf{w}(t)$ whose length is the vocabulary size, and with a "1" in a position uniquely associated with the word, and "0" elsewhere. $\mathbf{w}(t)$ and $\mathbf{s}(t)$ are concatenated to predict an output distribution over words, $\mathbf{y}(t)$. Updating is done with two weight matrices $\mathbf{u}$ and $\mathbf{v}$ and nonlinear functions $f()$ and $g()$ (Mikolov et al., 2011b):

$$\mathbf{x}(t) = [\mathbf{w}(t)^T \mathbf{s}(t-1)^T]^T$$

$$s_j(t) = f(\sum_i x_i(t) u_{ji})$$

$$y_k(t) = g(\sum_j s_j(t) v_{kj})$$

with $f()$ being a sigmoid and $g()$ a softmax:

$$f(x) = \frac{1}{1+\exp(-z)}, g(z_m) = \frac{\exp(z_m)}{\sum_k \exp(z_k)}$$

The output $\mathbf{y}(t)$ is a probability distribution over words, and the parameters $\mathbf{u}$ and $\mathbf{v}$ are trained with back-propagation to minimize the Kullback-Leibler (KL) divergence between the predicted and observed distributions. Because of the recurrent connections, this model is similar to a nonlinear infinite impulse response (IIR) filter, and has the potential to model long span dependencies. Theoretical considerations (Bengio et al., 1994) indicate that for many problems, this may not be possible, but in practice it is an empirical question.

4 Sentence Completion via Latent Semantic Analysis

Latent Semantic Analysis (LSA) (Deerwester et al., 1990) is a widely used method for representing words and documents in a low dimensional vector space. The method is based on applying singular value decomposition (SVD) to a matrix W representing the occurrence of words in documents. SVD results in an approximation of W by the product of three matrices, one in which each word is represented as a low-dimensional vector, one in which each document is represented as a low dimensional vector, and a diagonal scaling matrix. The similarity between two words can then be quantified as the cosine-similarity between their respective scaled vectors, and document similarity can be measured likewise. It has been used in numerous tasks, ranging from information retrieval (Deerwester et al., 1990) to speech recognition (Bellegarda, 2000; Coccaro and Jurafsky, 1998).

To perform LSA, one proceeds as follows. The input is a collection of n documents which are expressed in terms of words from a vocabulary of size m. These documents may be actual documents such as newspaper articles, or simply as in our case notional documents such as sentences. Next, a m x n matrix W is formed. At its simplest, the $ij^{\texttt{th}}$ entry contains the number of times word i has occurred in document j - its *term frequency* or TF value. More conventionally, the entry is weighted by some notion of the importance of word i, for example the negative logarithm of the fraction of documents that contain it, resulting in a TF-IDF weighting (Salton et al., 1975). Finally, to obtain a subspace representation of dimension d, W is decomposed as

$$W \approx USV^T$$

where U is m x d, V^T is d x n, and S is a d x d diagonal matrix. In applications, $d << n$ and $d << m$; for example one might have a $50,000$ word vocabulary and $1,000,000$ documents and use a 300 dimensional subspace representation.

An important property of SVD is that the rows of US - which represents the words - behave similarly to the original rows of W, in the sense that the cosine similarity between two rows in US approximates the cosine similarity between the corre-

sponding rows in W. Cosine similarity is defined as $\texttt{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$.

4.1 Total Word Similarity

Perhaps the simplest way of doing sentence completion with LSA is to compute the total similarity of a potential answer a with the rest of the words in the sentence $\mathcal{S}$, and to choose the most related option. We define the total similarity as:

$$\texttt{totsim}(a, \mathcal{S}) = \sum_{w \in \mathcal{S}} \texttt{sim}(a, w)$$

When the completion requires two words, total similarity is the sum of the contributions for both words. This is our baseline method for using LSA, and one of the best methods we have found.

4.2 Sentence Reconstruction

Recall that LSA approximates a weighted word-document matrix W as the product of low rank matrices U and V along with a scaling matrix S: $W \approx USV^T$. Using singular value decomposition, this is done so as to minimize the mean square reconstruction error $\sum_{ij} Q_{ij}^2$ where $Q = W - USV^T$. From the basic definition of LSA, each column of W (representing a document) is represented as

$$W_j = USV_j^T, \tag{1}$$

that is, as a linear combination of the set of basis functions formed by the columns of US, with the combination weights specified in V_j^T. When a new document is presented, it is also possible to represent it in terms of the same basis vectors. Moreover, we may take the reconstruction error induced by this representation to be a measure of how consistent the new document is with the original set of documents used to determine U S and V (Bellegarda, 2000).

It remains to represent a new document in terms of the LSA bases. This is done as follows (Deerwester et al., 1990; Bellegarda, 2000), again with the objective of minimizing the reconstruction error. First, note that since U is column-orthonormal, (1) implies that

$$V_j = W_j^T US^{-1} \tag{2}$$

Thus, if we notionally index a new document by p, we proceed by forming a new column (document) vector W_p using the standard term-weighting, and then find its LSA-space representation V_p using (2). We can evaluate the reconstruction quality by inserting the result in (1). The reconstruction error is then

$$||(UU^T - I)W_p||^2$$

Note that if all the dimensions are retained, the reconstruction error is zero; in the case that only the highest singular vectors are used, however, it is not. Due to the fact that the sentences vary in length we choose the number of retained singular vectors as a fraction f of the sentence length. If the answer has n words we use the top nf components. In practice, a f of 1.2 was selected on the basis of development set results.

4.3 A LSA N-gram Language Model

In the context of speech recognition, LSA has been combined with classical n-gram language models in (Coccaro and Jurafsky, 1998; Bellegarda, 2000). The crux of this idea is to interpolate an n-gram language model probability with one based on LSA, with the intuition that the standard n-gram model will do a good job predicting function words, and the LSA model will do a good job on words predicted by their long-span context. This logic makes sense for the sentence completion task as well, motivating us to evaluate it.

To do this, we adopt the procedure of (Coccaro and Jurafsky, 1998), using linear interpolation between the n-gram and LSA probabilities:

$$p(w|history) = \alpha p_{ng}(w|history) + (1 - \alpha) p_{lsa}(w|history)$$

The probability of a word given its history is computed by the LSA model in the following way. Let h be the sum of all the LSA word vectors in the history. Let m be the smallest cosine similarity between h and any word in the vocabulary V: $m = \min_{w \in V} \texttt{sim}(h, w)$. The probability of a word w in the context of history h is given by

$$P_{lsa}(w|h) = \frac{\texttt{sim}(h, w) - m}{\sum_{q \in V} (\texttt{sim}(h, q) - m)}$$

Since similarity can be negative, subtracting the minimum (m) ensures that all the estimated probabilities are between 0 and 1.

4.4 Improving Efficiency and Expressiveness

Given the basic framework described above, a number of enhancements are possible. In terms of efficiency, recall that it is necessary to perform SVD on a term-document matrix. The data we used was grouped into paragraph "documents," of which there were over 27 million, with 2.6 million unique words. While the resulting matrix is highly sparse, it is nevertheless impractical to perform SVD. We overcome this difficulty in two ways. First, we restrict the set of documents used to those which are "relevant" to a given test set. This is done by requiring that a document contain at least one of the potential answer-words. Secondly, we restrict the vocabulary to the set of words present in the test set. For the sentence-reconstruction method of Section 4.2, we have found it convenient to do data selection *per-sentence*.

To enhance the expressive power of LSA, the term vocabulary can be expanded from unigrams to bigrams or trigrams of words, thus adding information about word ordering. This was also used in the reconstruction technique.

5 Experimental Results

5.1 Data Resources

We present results with two datasets. The first is taken from *11 Practice Tests for the SAT & PSAT 2011 Edition* (Princeton-Review, 2010). This book contains eleven practice tests, and we used all the sentence completion questions in the first five tests as a development set, and all the questions in the last six tests as the test set. This resulted in sets with 95 and 108 questions respectively. Additionally, we report results on the recently released *MSR Sentence Completion Challenge* (Zweig and Burges, 2011). This consists of a set of $1,040$ sentence completion questions based on sentences occurring in five Conan Doyle Sherlock Holmes novels, and is identical in format to the SAT questions. Due to the source of this data, we refer to it as the *Holmes data*.

To train models, we have experimented with a variety of data sources. Since there is no publically available collection of SAT questions suitable to training, our methods have all relied on unsupervised data. Early on, we ran a set of experiments to determine the relevance of different types of data. Thinking that data from an encyclopedia might be useful, we evaluated an electronic version of the 2003 Encarta encyclopedia, which has approximately 29M words. Along similar lines, we used a collection of Wikipedia articles consisting of 709M words. This data is the entire Wikipedia as of January 2011, broken down into sentences, with filtering to remove sentences consisting of URLs and Wiki author comments. Finally, we used a commercial newspaper dataset consisting of all the Los Angeles Times data from 1985 to 2002, containing about 1.1B words. These data sources were evaluated using the baseline n-gram LM approach of Section 3.1. Initial experiments indicated that that the Los Angeles Times data is best suited to this task (see Table 1), and our SAT experiments use this source. For the MSR Sentence Completion data, we obtained the training data specified in (Zweig and Burges, 2011), consisting of approximately 500 19th-century novels available from Project Gutenberg, and comprising 48M words.

Data	Dev % Correct	Test % Correct
Encarta	26	33
Wikipedia	32	31
LA Times	39	42

Table 1: Effectiveness of different types of training data.

5.2 Human Performance

To provide human benchmark performance, we asked six native speaking high school students and five graduate students to answer the questions on the development set. The high-schoolers attained 87% accuracy and the graduate students 95%. Zweig and Burges (2011) cite a human performance of 91% on the Holmes data. Statistics from a large cross-section of the population are not available. As a further point of comparison, we note that chance performance is 20%.

5.3 Language Modeling Results

Table 2 summarizes our language modeling results on the SAT data. With the exception of the baseline backoff n-gram model, these techniques were too computationally expensive to utilize the full Los Angeles Times corpus. Instead, as with LSA, a "relevant" corpus was selected of the sentences which contain at least one answer option from either the

Method	**Data (Dev / Test)**	**Dev**	**Test**
3-gram GT	1.1B / 1.1B	39%	42%
Model M	193M / 236M	35	41
RNN	36M / 44M	37	42
LSA-LM	293M / 358 M	48	44

Table 2: Performance of language modeling methods on SAT questions.

Method	**Dev ppl**	**Dev**	**Test ppl**	**Test**
3-gram GT	195	36%	190	44%
Model M	178	36	175	42
RNN	147	37	144	42

Table 3: Performance of language modeling methods using identical training data and vocabularies.

development or test set. Separate subsets were made for development and test data. This data was further sub-sampled to obtain the training set sizes indicated in the second column. For the LSA-LM, an interpolation weight of 0.1 was used for the LSA score, determined through optimization on the development set. We see from this table that the language models perform similarly and achieve just above 40% on the test set.

To make a more controlled comparison that normalizes for the amount of training data, we have trained Model M, and the Good-Turing model on the same data subset as the RNN, and with the same vocabulary. In Table 3, we present perplexity results on a held-out set of dev/test-relevant Los Angeles Times data, and performance on the actual SAT questions. Two things are notable. First, the recurrent neural net has dramatically lower perplexity than the other methods. This is consistent with results in (Mikolov et al., 2011a). Secondly, despite the differences in perplexity, the methods show little difference on SAT performance. Because Model M was not better, only uses n-gram context, and was used in the construction of the Holmes data (Zweig and Burges, 2011), we do not consider it further.

5.4 LSA Results

Table 4 presents results for the methods of Sections 4.1 and 4.2. Of all the methods in isolation, the simple approach of Section 4.1 - to use the total cosine similarity between a potential answer and the other words in the sentence - has performed best. The approach of using reconstruction error performed very well on the development set, but unremarkably on the test set.

Method	**Dev**	**Test**
Total Word Similarity	46%	46%
Reconstruction Error	53	41

Table 4: SAT performance of LSA based methods.

Method	**Test**
3-input LSA	46%
LSA + Good-Turing LM	53
LSA + Good-Turing LM + RNN	52

Table 5: SAT test set accuracy with combined methods.

5.5 Combination Results

A well-known trick for obtaining best results from a machine learning system is to combine a set of diverse methods into a single ensemble (Dietterich, 2000). We use ensembles to get the highest accuracy on both of our data sets.

We use a simple linear combination of the outputs of the other models discussed in this paper. For the LSA model, the linear combination has three inputs: the total word similarity, the cosine similarity between the sum of the answer word vectors and the sum of the rest of sentence's word vectors, and the number of out-of-vocabulary terms in the answer. Each additional language model beyond LSA contributes an additional input: the probability of the sentence under that language model.

We train the parameters of the linear combination on the SAT development set. The training minimizes a loss function of pairs of answers: one correct and one incorrect fill-in from the same question. We use the RankNet loss function (Burges et al., 2005):

$$\min_{\vec{w}} f(\vec{w} \cdot (\vec{x} - \vec{y})) + \lambda ||\vec{w}||^2$$

where $\vec{x}$ are the input features for the incorrect answer, $\vec{y}$ are the features for the correct answer, $\vec{w}$ are the weights for the combination, and $f(z) = \log(1 + \exp(z))$. We tune the regularizer via 5-fold cross validation, and minimize the loss using L-BFGS (Nocedal and Wright, 2006). The results on the SAT test set for combining various models are shown in Table 5.

5.6 Holmes Data Results

To measure the robustness of our approaches, we have applied them to the MSR Sentence Completion set (Zweig and Burges, 2011), termed the *Holmes data*. In Table 6, we present the results on this set, along with the comparable SAT results. Note that the latter are derived from models trained with the Los Angeles Times data, while the Holmes results are derived from models trained with 19th-century novels. We see from this table that the results are similar across the two tasks. The best performing single model is LSA total word similarity.

For the Holmes data, combining the models outperforms any single model. We train the linear combination function via 5-fold cross-validation: the model is trained five times, each time on $3/5$ of the data, the regularization tuned on $1/5$ of the data, and tested on $1/5$. The test results are pooled across all 5 folds and are shown in Table 6. In this case, the best combination is to blend LSA, the Good-Turing language model, and the recurrent neural network.

Method	**SAT**	**Holmes**
Chance	20%	20%
GT N-gram LM	42	39
RNN	42	45
LSA Total Similarity	46	49
Reconstruction Error	41	41
LSA-LM	44	42
Combination	**53**	**52**
Human	87 to 95	91

Table 6: Performance of methods on the MSR Sentence Completion Challenge, contrasted with SAT test set.

6 Discussion

To verify that the differences in accuracy between the different algorithms are not statistical flukes, we perform a statistical significance test on the outputs of each algorithm. We use McNemar's test, which is a matched test between two classifiers (Dietterich, 1998). We use the False Discovery Rate method (Benjamini and Hochberg, 1995) to control the false positive rate caused by multiple tests. If we allow 2% of our tests to yield incorrectly false results, then for the SAT data, the combination of the Good-Turing smoothed language model with an LSA-based global similarity model (52% accuracy) is better that the baseline alone (42% accuracy).

Secondly, for the Holmes data, we can state that LSA total similarity beats the recurrent neural network, which in turn is better than the baseline n-gram model. The combination of all three is significantly better than any of the individual models.

To better understand the system performance and gain insight into ways of improving it, we have examined the system's errors. Encouragingly, one-third of the errors involve single-word questions which test the dictionary definition of a word. This is done either by stating the definition, or providing a stereotypical use of the word. An example of the first case is: "Great artists are often *prophetic (visual)*: they perceive what we cannot and anticipate the future long before we do." (The system's incorrect answer is in parentheses.) An example of the second is: "One cannot help but be moved by Theresa's *heartrending (therapeutic)* struggle to overcome a devastating and debilitating accident."

At the other end of the difficulty spectrum are questions involving world knowledge and/or logical implications. An example requiring both is, "Many fear that the *ratification (withdrawal)* of more lenient tobacco advertising could be detrimental to public health." About 40% of the errors require this sort of general knowledge to resolve. Based on our analysis, we believe that future research could profitably exploit the structured information present in a dictionary. However, the ability to identify and manipulate logical relationships and embed world knowledge in a manner amenable to logical manipulation may be necessary for a full solution. It is an interesting research question if this could be done implicitly with a machine learning technique, for example recurrent or recursive neural networks.

7 Conclusion

In this paper we have investigated methods for answering sentence-completion questions. These questions are intriguing because they probe the ability to distinguish semantically coherent sentences from incoherent ones, and yet involve no more context than the single sentence. We find that both local n-gram information and an LSA-based global coherence model do significantly better than chance, and that they can be effectively combined.

References

J. Bellegarda. 2000. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8).

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157 –166.

Y. Benjamini and Y. Hochberg. 1995. Controlling the fase discovery rate: a practical and powerful approach to multiple testing. *J. Royal Statistical Society* **B**, 53(1):289–300.

C. Burges, T. Shaked., E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. 2005. Learning to rank using gradient descent. In *Proc. ICML*, pages 89–96.

Eugene Charniak, Yasemin Altun, Rodrigo de Salvo Braz, Benjamin Garrett, Margaret Kosmala, Tomer Moscovich, Lixin Pang, Changhee Pyo, Ye Sun, Wei Wy, Zhongfa Yang, Shawn Zeller, and Lisa Zorn. 2000. Reading comprehension programs in a statistical-language-processing class. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems - Volume 6*, ANLP/NAACL-ReadingComp '00, pages 1–5. Association for Computational Linguistics.

Stanley Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–393.

S. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy. 2009. Scaling shrinkage-based language models. In *ASRU*.

S. Chen. 2009a. Performance prediction for exponential language models. In *NAACL-HLT*.

S. Chen. 2009b. Shrinking exponential language models. In *NAACL-HLT*.

P.R. Clarkson and R. Rosenfeld. 1997. Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings ESCA Eurospeech*, http://www.speech.cs.cmu.edu/SLM/toolkit.html.

N. Coccaro and D. Jurafsky. 1998. Towards better integration of semantic predictors in statistical language modeling. In *Proceedings, ICSLP*.

Bollegala D., Matsuo Y., and Ishizuka M. 2009. Measuring the similarity between implicit semantic relations from the web. In *World Wide Web Conference (WWW)*.

S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(96).

T.G. Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923.

T.G. Dietterich. 2000. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer-Verlag.

Educational-Testing-Service. 2011. https://satonlinecourse.collegeboard.com/sr/digital_assets/assessment/pdf/0833a611-0a43-10c2-0148-cc8c0087fb06-f.pdf.

A. Emami, S. Chen, A. Ittycheriah, H. Soltau, and B. Zhao. 2010. Decoding with shrinkage-based language models. In *Interspeech*.

Claudio Giuliano, Alfio Gliozzo, and Carlo Strapparava. 2007. Fbk-irst: Lexical substitution task exploiting domain and syntagmatic coherence. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 145–148, Stroudsburg, PA, USA. Association for Computational Linguistics.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. Unt: Subfinder: Combining knowledge sources for automatic lexical substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 410–413, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lynette Hirschman, Mark Light, Eric Breck, and John D. Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*.

Thomas Landauer and Susan Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review, 104(2)*, pages 211–240.

Iddo Lev, Bill MacCartney, Christopher D. Manning, and Roger Levy. 2004. Solving logic puzzles: from robust processing to precise semantics. In *Proceedings of the 2nd Workshop on Text Meaning and Interpretation*, pages 9–16. Association for Computational Linguistics.

Jarmasz M. and Szpakowicz S. 2003. Roget's thesaurus and semantic similarity. In *Recent Advances in Natural Language Processing (RANLP)*.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53.

Tomas Mikolov, Martin Karafiat, Jan Cernocky, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech 2010*.

Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukas Burget, and Jan Cernocky. 2011a. Empirical evaluation and combination of advanced language modeling techniques. In *Proceedings of Interspeech 2011*.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocky, and Sanjeev Khudanpur. 2011b. Extensions of recurrent neural network based language model. In *Proceedings of ICASSP 2011*.

Saif Mohammed, Bonnie Dorr, and Graeme Hirst. 2008. Computing word pair antonymy. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Saif M. Mohammed, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2011. Measuring degrees of semantic opposition. Technical report, National Research Council Canada.

Hwee Tou Ng, Leong Hwee Teo, and Jennifer Lai Pheng Kwan. 2000. A machine learning approach to answering questions for reading comprehension tests. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 124–132.

J. Nocedal and S. Wright. 2006. *Numerical Optimization*. Springer-Verlag.

Sebastian Pado and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics, 33 (2)*, pages 161–199.

Princeton-Review. 2010. *11 Practice Tests for the SAT & PSAT, 2011 Edition*. The Princeton Review.

Ellen Riloff and Michael Thelen. 2000. A rule-based question answering system for reading comprehension tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems - Volume 6*, ANLP/NAACL-ReadingComp '00, pages 13–19.

G. Salton, A. Wong, and C. S. Yang. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11).

Richard Socher, Cliff Chiung-Yu Lin, Andrew Y. Ng, and Christopher D. Manning. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 2011 International Conference on Machine Learning (ICML-2011)*.

Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating text with recurrent neural networks. In *Proceedings of the 2011 International Conference on Machine Learning (ICML-2011)*.

E. Terra and C. Clarke. 2003. Frequency estimates for statistical word similarity measures. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Peter Turney and Michael Littman. 2005. Corpus-based learning of analogies and semantic relations. *Machine Learning, 60 (1-3)*, pages 251–278.

Peter D. Turney, Michael L. Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Recent Advances in Natural Language Processing (RANLP)*.

Peter D. Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *European Conference on Machine Learning (ECML)*.

Peter Turney. 2008. A uniform approach to analogies, synonyms, antonyms, and associations. In *International Conference on Computational Linguistics (COLING)*.

T. Veale. 2004. Wordnet sits the sat: A knowledge-based approach to lexical analogy. In *European Conference on Artificial Intelligence (ECAI)*.

W. Wang, J. Auer, R. Parasuraman, I. Zubarev, D. Brandyberry, and M. P. Harper. 2000. A question answering system developed as a project in a natural language processing course. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems - Volume 6*, ANLP/NAACL-ReadingComp '00, pages 28–35.

Deniz Yuret. 2007. Ku: word sense disambiguation by substitution. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 207–213, Stroudsburg, PA, USA. Association for Computational Linguistics.

Geoffrey Zweig and Christopher J.C. Burges. 2011. The Microsoft Research sentence completion challenge. Technical Report MSR-TR-2011-129, Microsoft.