Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information

Yutaka Matsuo
National Institute of Advanced Industrial Science and Technology
Aomi 2-41-6, Koto-ku, Tokyo 135-0064, Japan
y.matsuo@carc.aist.go.jp

Mitsuru Ishizuka
University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-8656, Japan
ishizuka@miv.t.u-tokyo.ac.jp

Abstract

We present a new keyword extraction algorithm that applies to a single document without using a corpus. Frequent terms are extracted first, then a set of co-occurrence between each term and the frequent terms, i.e., occurrences in the same sentences, is generated. Co-occurrence distribution shows importance of a term in the document as follows. If probability distribution of co-occurrence between term a and the frequent terms is biased to a particular subset of frequent terms, then term a is likely to be a keyword. The degree of biases of distribution is measured by the χ^2-measure. Our algorithm shows comparable performance to *tfidf* without using a corpus.

Introduction

Keyword extraction[1] is an important technique for document retrieval, Web page retrieval, document clustering, summarization, text mining, and so on. By extracting appropriate keywords, we can choose easily which document to read or learn the relation among documents. A popular algorithm for indexing is the *tfidf* measure, which extracts keywords frequently that appear in a document, but don't appear frequently in the remainder of the corpus.

Recently, numerous documents have been made available electronically. Domain-independent keyword extraction, which does not require a large corpus, has many applications. For example, if one encounters a new Web page, one might like to know the contents quickly by some means, e.g., highlighting keywords. If one wants to know the main assertion of a paper at hand, one would want to have some keywords. In these cases, a keyword extraction without a corpus of the same kind of documents is very useful. Word count (Luhn 1957) is sometimes sufficient for document overview; however, a more powerful tool is desirable.

This paper explains a keyword extraction algorithm based solely on a single document. First, frequent terms[2] are extracted. Co-occurrences of a term and frequent terms are counted. If a term appears selectively with a particular subset of frequent terms, the term is likely to have an important meaning. The degree of bias of the co-occurrence distribution is measured by the χ^2-measure. We show that our keyword extraction performs well without the need for a corpus.

This paper is organized as follows. The next section describes our main idea of keyword extraction. We detail the algorithm, then evaluation and discussion are made. Finally, we summarize our contributions and conclude the paper.

Term Co-occurrence and Importance

A document consists of sentences. In this paper, a sentence is considered to be a set of words separated by a stop mark (".", "?" or "!"). Moreover, it includes a title of a document, a title of a section, and a caption. Two terms in a sentence are considered to co-occur once. That is, we see each sentence as a "basket," ignoring term order and grammatical information except when extracting word sequences.

We can obtain frequent terms by counting term frequencies. Let us take a very famous paper by Alan Turing (Turing 1950) as an example. Table 1 shows the top ten frequent terms (denoted as G) and the probability of occurrence, normalized so that the sum is to be 1 (i.e., normalized relative frequency). Next, a co-occurrence matrix is obtained by counting frequencies of pairwise term co-occurrence, as shown in Table 2. For example, term a and term b co-occur in 30 sentences in the document. Let N denote the number of different terms in the document. While the term co-occurrence matrix is an $N \times N$ symmetric matrix, Table 2 shows only a part of the whole – an $N \times 10$ matrix. We do not define diagonal components here.

Assuming that term w appears independently from frequent terms G, the distribution of co-occurrence of term w and the frequent terms is similar to the unconditional distribution of occurrence of the frequent terms shown in Table 1. Conversely, if term w has a semantic relation with a particular set of terms $g \in G$, co-occurrence of term w and g is greater than expected; the distribution is to be biased.

Figures 1 and 2 show co-occurrence probability distribution of some terms and the frequent terms. In the figures, unconditional distribution of frequent terms is shown as "un-

[1] A term "keyword extraction" is used in the context of text mining, for example (Rajman & Besancon 1998). A comparable research topic is called "automatic term recognition" in the context of computational linguistics and "automatic indexing" or "automatic keyword extraction" in the information retrieval research field.

[2] A term is a word or a word sequence. (We do not intend to limit the meaning in a terminological sense.) A word sequence is written as a phrase.

Table 1: Frequency and probability distribution.

Frequent term	a	b	c	d	e	f	g	h	i	j	Total
Frequency	203	63	44	44	39	36	35	33	30	28	555
Probability	0.366	0.114	0.079	0.079	0.070	0.065	0.063	0.059	0.054	0.050	1.0

a: *machine*, b: *computer*, c: *question*, d: *digital*, e: *answer*, f: *game*, g: *argument*, h: *make*, i: *state*, j: *number*

Table 2: A co-occurrence matrix.

	a	b	c	d	e	f	g	h	i	j	Total
a	–	30	26	19	18	12	12	17	22	9	165
b	30	–	5	50	6	11	1	3	2	3	111
c	26	5	–	4	23	7	0	2	0	0	67
d	19	50	4	–	3	7	1	1	0	4	89
e	18	6	23	3	–	7	1	2	1	0	61
f	12	11	7	7	7	–	2	4	0	0	50
g	12	1	0	1	1	2	–	5	1	0	23
h	17	3	2	1	2	4	5	–	0	0	34
i	22	2	0	0	1	0	1	0	–	7	33
j	9	3	0	4	0	0	0	0	7	–	23
...	...	...	...	...	...	...	...	...	...	...	...
u	6	5	5	3	3	18	2	2	1	0	45
v	13	40	4	35	3	6	1	0	0	2	104
w	11	2	2	1	1	0	1	4	0	0	22
x	17	3	2	1	2	4	5	0	0	0	34

u: *imitation*, v: *digital computer*, w:*kind*, x:*make*

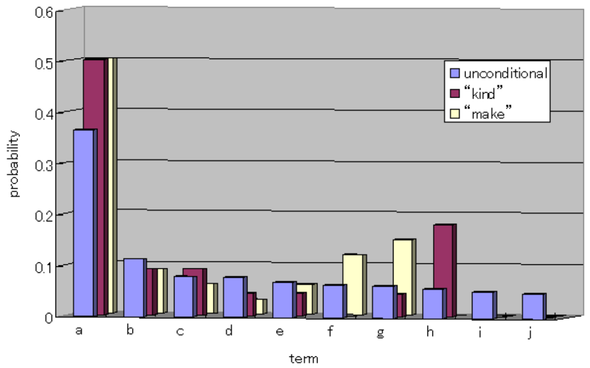

Figure 1: Co-occurrence probability distribution of the terms *"kind"*, *"make"*, and frequent terms.

conditional". A general term such as 'kind" or "make" is used relatively impartially with each frequent term, while a term such as "imitation" or "digital computer" shows co-occurrence especially with particular terms. These biases are derived from either semantic, lexical, or other relations of two terms. Thus, a term with co-occurrence biases may have an important meaning in a document. In this example, "imitation" and "digital computer" are important terms, as we all know: In this paper, Turing proposed an "imitation game" to replace the question "Can machines think?"

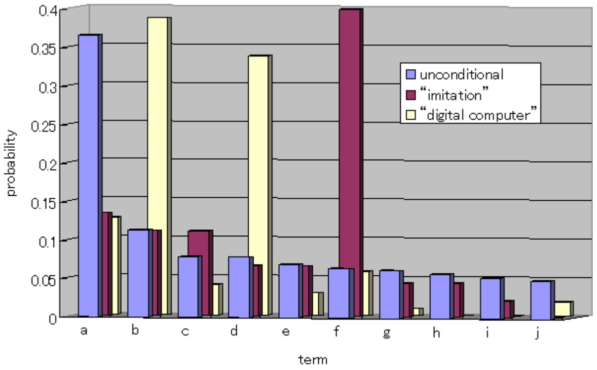

Figure 2: Co-occurrence probability distribution of the terms *"imitation"*, *"digital computer"*, and frequent terms.

Therefore, the degree of biases of co-occurrence can be used as a surrogate of term importance. However, if term frequency is small, the degree of biases is not reliable. For example, assume term w_1 appears only once and co-occurs only with term a once (probability 1.0). On the other extreme, assume term w_2 appears 100 times and co-occurs only with term a 100 times (with probability 1.0). Intuitively, w_2 seems more reliably biased. In order to evaluate statistical significance of biases, we use the χ^2 test, which is very common for evaluating biases between expected frequencies and observed frequencies. For each term, frequency of co-occurrence with the frequent terms is regarded as a sample value; a null hypothesis is that "occurrence of frequent terms G is independent from occurrence of term w," which we expect to reject.

We denote the unconditional probability of a frequent term $g \in G$ as the expected probability p_g and the total number of co-occurrence of term w and frequent terms G as n_w. Frequency of co-occurrence of term w and term g is written as $freq(w, g)$. The statistical value of χ^2 is defined as

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g}. \qquad (1)$$

If $\chi^2(w) > \chi^2_\alpha$, the null hypothesis is rejected with significance level α. The term $n_w p_g$ represents the expected frequency of co-occurrence; and $(freq(w, g) - n_w p_g)$ represents the difference between expected and observed frequencies. Therefore, large $\chi^2(w)$ indicates that co-occurrence of term w shows strong bias. In this paper, we use the χ^2-measure as an index of biases, not for tests of hypotheses.

Table 3 shows terms with high χ^2 values and ones with low χ^2 values in the Turing's paper. Generally, terms with large χ^2 are relatively important in the document; terms with small χ^2 are relatively trivial.

In summary, our algorithm first extracts frequent terms as a "standard"; then it extracts terms with high deviation from the standard as keywords.

Algorithm Description and Improvement

This section details precise algorithm description and algorithm improvement based on preliminary experiments.

Calculation of χ^2 values

A document consists of sentences of various lengths. If a term appears in a long sentence, it is likely to co-occur with

Table 3: Terms with high χ^2 value.

Rank	χ^2	Term	Frequency
1	593.7	digital computer	31
2	179.3	imitation game	16
3	163.1	future	4
4	161.3	question	44
5	152.8	internal	3
6	143.5	answer	39
7	142.8	input signal	3
8	137.7	moment	2
9	130.7	play	8
10	123.0	output	15
⋮	⋮	⋮	⋮
553	0.8	Mr.	2
554	0.8	sympathetic	2
555	0.7	leg	2
556	0.7	chess	2
557	0.6	Pickwick	2
558	0.6	scan	2
559	0.3	worse	2
560	0.1	eye	2

(We set the top ten frequent terms as G.)

many terms; if a term appears in a short sentence, it is less likely to co-occur with other terms. We consider the length of each sentence and revise our definitions. We denote

- p_g as (the sum of the total number of terms in sentences where g appears) divided by (the total number of terms in the document),
- n_w as the total number of terms in sentences where w appears.

Again $n_w p_g$ represents the expected frequency of co-occurrence. However, its value becomes more sophisticated.

A term co-occurring with a particular term $g \in G$ has a high χ^2 value. However, these terms are sometimes adjuncts of term g and not important terms. For example, in Table 3, a term "future" or "internal" co-occurs selectively with the frequent term "state," because these terms are used in the form of "future state" and "internal state." Though χ^2 values for these terms are high, "future" and "internal" themselves are not important. Assuming the "state" is not a frequent term, χ^2 values of these terms diminish rapidly.

We use the following function to measure robustness of bias values; it subtracts the maximal term from the χ^2 value,

$$\chi'^2(w) = \chi^2(w) - \max_{g \in G} \left\{ \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g} \right\}. \quad (2)$$

Clustering of Terms

A co-occurrence matrix is originally an $N \times N$ matrix, where columns corresponding to frequent terms are extracted for calculation. We ignore the remaining of columns, i.e., co-occurrence with low frequency terms, because it is difficult to estimate precise probability of occurrence for low frequency terms.

To improve extracted keyword quality, it is very important to select the proper set of columns from a co-occurrence matrix. The set of columns is preferably orthogonal; assuming that terms g_1 and g_2 appear together very often, co-occurrence of terms w and g_1 might imply the co-occurrence of w and g_2. Thus, term w will have a high χ^2 value; this is very problematic.

Table 4: Two transposed columns.

	a	b	c	d	e	f	g	h	i	j	. . .
c	**26**	**5**	—	**4**	*23*	**7**	**0**	**2**	**0**	**0**	. . .
e	**18**	**6**	*23*	**3**	—	**7**	**1**	**2**	**1**	**0**	. . .

Table 5: Clustering of the top 49 frequent terms.

C1:	game, imitation, imitation game, play, programme				
C2:	system, rules, result, important				
C3:	computer, digital, digital computer				
C4:	behaviour, random, law				
C5:	capacity, storage,	C6:	question, answer		
. . .	. . .				
C26:	human,	C27:	state,	C28:	learn

It is straightforward to extract an orthogonal set of columns, however, to prevent the matrix from becoming too sparse, we will cluster terms (i.e., columns).

Many studies address term clustering. Two major approaches (Hofmann & Puzicha 1998) are:

Similarity-based clustering If terms w_1 and w_2 have similar distribution of co-occurrence with other terms, w_1 and w_2 are considered to be the same cluster.

Pairwise clustering If terms w_1 and w_2 co-occur frequently, w_1 and w_2 are considered to be the same cluster.

Table 4 shows an example of two (transposed) columns extracted from a co-occurrence matrix. Similarity-based clustering centers upon boldface figures and pairwise clustering focuses on italic figures.

By similarity-based clustering, terms with the same role, e.g., "Monday," "Tuesday," ..., or "build," "establish," and "found" are clustered (Pereira, Tishby, & Lee 1993). Through our preliminary experiment, when applied to a single document, similarity-based clustering groups paraphrases, and a phrase and its component (e.g., "digital computer" and "computer"). Similarity of two distributions is measured statistically by Kullback-Leibler divergence or Jensen-Shannon divergence (Dagan, Lee, & Pereira 1999).

On the other hand, pairwise clustering yields relevant terms in the same cluster: "doctor," "nurse," and "hospital" (Tanaka & Iwasaki 1996). A frequency of co-occurrence or mutual information can be used to measure the degree of relevance (Church & Hanks 1990; Dunning 1993).

Our algorithm uses both types of clustering. First we cluster terms by a similarity measure (using Jensen-Shannon divergence); subsequently, we apply pairwise clustering (using mutual information). Table 5 shows an example of term clustering. Proper clustering of frequent terms results in an appropriate χ^2 value for each term.[3]

Below, co-occurrence of a term and a cluster implies co-occurrence of the term and any term in the cluster.

Algorithm

The algorithm is shown as follows. Thresholds are determined by preliminary experiments.

[3] Here we don't take the size of the cluster into account. Balancing the clusters may improve the algorithm performance.

Table 6: Improved results of terms with high χ^2 value.

Rank	χ^2	Term	Frequency
1	380.4	digital computer	63
2	259.7	storage capacity	11
3	202.5	imitation game	16
4	174.4	machine	203
5	132.2	human mind	2
6	94.1	universality	6
7	93.7	logic	10
8	82.0	property	11
9	77.1	mimic	7
10	77.0	discrete-state machine	17

1. Preprocessing: Stem words by Porter algorithm (Porter 1980) and extract phrases based on the APRIORI algorithm (Fürnkranz 1998). Discard stop words included in stop list used in SMART system (Salton 1988).[4]
2. Selection of frequent terms: Select the top frequent terms up to 30% of the number of running terms, N_{total}.
3. Clustering frequent terms: Cluster a pair of terms whose Jensen-Shannon divergence is above the threshold ($0.95 \times \log 2$). Cluster a pair of terms whose mutual information is above the threshold ($\log(2.0)$). The obtained clusters are denoted as C.
4. Calculation of expected probability: Count the number of terms co-occurring with $c \in C$, denoted as n_c, to yield the expected probability $p_c = n_c/N_{total}$.
5. Calculation of χ'^2 value: For each term w, count co-occurrence frequency with $c \in C$, denoted as $freq(w, c)$. Count the total number of terms in the sentences including w, denoted as n_w. Calculate χ'^2 value following (2).
6. Output keywords: Show a given number of terms having the largest χ'^2 value.

Table 6 shows the result for Turing's paper. Important terms are extracted regardless of their frequencies.

Evaluation

For information retrieval, index terms are evaluated by their retrieval performance, namely recall and precision. However, we claim that our algorithm is useful when a corpus is not available due to cost or time to collect documents, or in a situation where document collection is infeasible.

The experiment was participated by 20 authors of technical papers in artificial intelligence research. For each author, we showed keywords of his/her paper by *tf*(term frequency), *tfidf*[5], *KeyGraph*[6] (Ohsawa, Benson, & Yachida 1998), and our algorithm. All these methods are equally equipped with word stem, elimination of stop words, and extraction of phrases. The top 15 terms by each method were extracted, gathered, and shuffled. Then, the authors were asked to check terms which they think are important in the paper.[7] Precision can be calculated by the ratio of the checked terms to 15 terms derived by each method. Furthermore, the authors were asked to select five (or more) terms which they thought were indispensable for the paper. Coverage of each method was calculated by taking the ratio of the indispensable terms included in the 15 terms to all the indispensable terms.[8]

Table 7: Precision and coverage for 20 technical papers.

	tf	*KeyGraph*	**ours**	*tfidf*
Precision	0.53	0.42	**0.51**	0.55
Coverage	0.48	0.44	**0.62**	0.61
Frequency index	28.6	17.3	**11.5**	18.1

Table 8: Results with respect to phrases.

	tf	*KeyGraph*	**ours**	*tfidf*
Ratio of phrases	0.11	0.14	**0.33**	0.33
Precision w/o phrases	0.42	0.36	**0.42**	0.45
Recall w/o phrases	0.39	0.36	**0.46**	0.54

Results are shown in Table 7. For each method, precision was around 0.5. However, coverage using our method exceeds that of *tf* and *KeyGraph* and is comparable to that of *tfidf*; both *tf* and *tfidf* selected terms which appeared frequently in the document (although *tfidf* considers frequencies in other documents). On the other hand, our method can extract keywords even if they do not appear frequently. The frequency index in the table shows average frequency of the top 15 terms. Terms extracted by *tf* appear about 28.6 times, on average, while terms by our method appear only 11.5 times. Therefore, our method can detect "hidden" keywords. We can use χ^2 value as a priority criterion for keywords because precision of the top 10 terms by our method is 0.52, that of the top 5 is 0.60, while that of the top 2 is as high as 0.72. Though our method detects keywords consisting of two or more words well, it is still nearly comparable to *tfidf* if we discard such phrases, as shown in Table 8.

Computational time of our method is shown in Figure 3. The system is implemented in C++ on a Linux OS, Celeron 333MHz CPU machine. Computational time increases approximately linearly with respect to the number of terms; the process completes itself in a few seconds if the given number of terms is less than 20,000.

Discussion and Related Works

Co-occurrence has long attracted interest in computational linguistics. (Pereira, Tishby, & Lee 1993) clustered terms according to their distribution in particular syntactic contexts. (Tanaka & Iwasaki 1996) uses co-occurrence matrices of two languages to translate an ambiguous term. From

[4] In this paper, we use both nouns and verbs because verbs or verb+noun are sometimes important to illustrate the content of the document. Of course, we can apply our algorithm only to nouns.

[5] The corpus is 166 papers in JAIR (Journal of Artificial Intelligence Research) from Vol. 1 in 1993 to Vol. 14 in 2001. The *idf* is defined by $\log(D/df(w)) + 1$, where D is the number of all documents and $df(w)$ is the number of documents including w.

[6] This term-weighting algorithm, which is recently used to analyze a variety of data in the context of *Chance Discovery*,requires only a single document.

[7] Keywords are sometimes attached to a paper; however, they are not defined in a consistent way. Therefore, we employ author-based evaluation.

[8] It is desirable to have the indispensable term list beforehand. However, it is very demanding for authors to provide a keyword list without seeing a term list. In our experiment, we allowed authors to add any terms in the paper to include in the indespensable term list (even if they were not derived by any of the methods.).

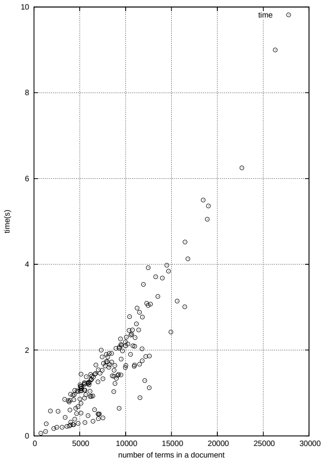

Figure 3: Number of total terms and computational time.

probabilistic points of view, (Dagan, Lee, & Pereira 1999) describes a method for estimating probability of previously unseen word combinations.

Weighting a term by occurrence dates back to the 1950s and the study by Luhn (1957). More elaborate measures of term occurrence have been developed (Sparck-Jones 1972; Noreault, McGill, & Koll 1977) by essentially counting term frequencies. (Kageura & Umino 1996) summarized five groups of weighting measure: (i) a word which appears in a document is likely to be an index term; (ii) a word which appears frequently in a document is likely to be an index term; (iii) a word which appears only in a limited number of documents is likely to be an index term for these documents; (iv) a word which appears relatively more frequently in a document than in the whole database is likely to be an index term for that document; (v) a word which shows a specific distributional characteristic in the database is likely to be an index term for the database. Our algorithm corresponds to approach (v). (Nagao, Mizutani, & Ikeda 1976) used χ^2 value to calculate weight of words using a corpus. Our method uses a co-occurrence matrix instead of a corpus, enabling keyword extraction using only the document itself.

In the context of text mining, to discover keywords or relation of keywords are important topics (Feldman *et al.* 1998; Rajman & Besancon 1998). The general purpose of knowledge discovery is to extract implicit, previously unknown, and potentially useful information from data. Our algorithm can be considered as a text mining tool in that it extracts important terms even if they are rare.

Conclusion

In this paper, we developed an algorithm to extract keywords from a single document. Main advantages of our method are its simplicity without requiring use of a corpus and its high performance comparable to *tfidf*. As more electronic documents become available, we believe our method will be useful in many applications, especially for domain-independent keyword extraction.

References

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16(1):22.

Dagan, I.; Lee, L.; and Pereira, F. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning* 34(1):43.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1):61.

Feldman, R.; Fresko, M.; Kinar, Y.; Lindell, Y.; Liphstat, O.; Rajman, M.; Schler, Y.; and Zamir, O. 1998. Text mining at the term level. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, 65.

Fürnkranz, J. 1998. A study using n-grams features for text categorization. Technical Report OEFAI-TR-98-30, Austrian Research Institute for Artificial Intelligence.

Hofmann, T., and Puzicha, J. 1998. Statistical models for co-occurrence data. Technical Report AIM-1625, Massachusetts Institute of Technology.

Kageura, K., and Umino, B. 1996. Methods of automatic term recognition. *Terminology* 3(2):259.

Luhn, H. P. 1957. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development* 1(4):390.

Nagao, M.; Mizutani, M.; and Ikeda, H. 1976. An automated method of the extraction of important words from Japanese scientific documents. *Transactions of Information Processing Society of Japan* 17(2):110.

Noreault, T.; McGill, M.; and Koll, M. B. 1977. *A Performance Evaluation of Similarity Measure, Document Term Weighting Schemes and Representations in a Boolean Environment*. London: Butterworths.

Ohsawa, Y.; Benson, N. E.; and Yachida, M. 1998. KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proceedings of the Advanced Digital Library Conference*.

Pereira, F.; Tishby, N.; and Lee, L. 1993. Distributional clustering of English words. In *Proceedings of the 31th Meeting of the Association for Computational Linguistics*, 183–190.

Porter, M. F. 1980. An algorithm for suffix stripping. *Program* 14(3):130.

Rajman, M., and Besancon, R. 1998. Text mining – knowledge extraction from unstructured textual data. In *Proceedings of the 6th Conference of International Federation of Classification Societies*.

Salton, G. 1988. *Automatic Text Processing*. Addison-Wesley.

Sparck-Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28(5):111.

Tanaka, K., and Iwasaki, H. 1996. Extraction of lexical translations from non-aligned corpora. In *Proceedings of the 16th International Conference on Computational Linguistics*, 580.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind* 59:433.