

# 英語読解問題の自動解法

Automatic Solvers for English Reading Comprehension Problems

佐藤敦美 佐々木裕  
Atsumi Sato Yutaka Sasaki

豊田工業大学  
Toyota Technological Institute

## 1. はじめに

今日、情報化社会が日々進歩していくうえで、自然言語処理技術がより多くの分野で用いられるようになってきている。その利用範囲は医療、検索エンジン、翻訳、音声認識等、多岐にわたり、そのどれもが言語の持つ意味を解析する処理工程を含んでいる。しかし、人間が実際に使う言語には多くの前提条件、推定が含まれておりシステムに処理させるのは非常に難しいとされている。意味解析は自然言語処理を進展させていく上で最も重大な課題であり、現在問題となっているボトルネックを解消するために必要不可欠である。そこで、本研究では意味解析が特に必要とされる読解問題を通じて、意味解析に効果的な手法を検討する。

本研究では日本でも英語能力の評価指標として多く取り入れられている TOEIC の問題をシステムに解答させるための、効果的な意味解析手法を検討する。TOEIC の問題は様々な出題形式を取り入れているが今回は読解問題を解くシステムについて取り扱うため、Part 7 の問題のみを対象とした。

## 2. 検討手法

解答方法として、選択肢の単語と問題中の単語について単語間類似度、及びパターンマッチングを用いる。単語間類似度による意味的なマッチングを行うため、概念辞書である WordNet[1]を用いた。

WordNet は与えられた単語と品詞(名詞、動詞、形容詞、副詞)から単語の原形、複数の同義語分類番号(Offset)、単語の持つ意味ごとの同義語、例文などを取得するものである。意味的な解析を行う上でまず単語が同一を Offset に属しているのか、いないのかという二値判定から正答率を得る。続いて WordNet::Similarity[2]を用いて単語間類似度からみた正答率を検証した。

WordNet::Similarity にはあらかじめ、いくつかのメソッドが組み立てられており、それぞれ異なった方法で単語間類似度を算出する。今回用いた単語間類似度のメソッドは以下の7つである。

**Jiang&Conrath (JCN) [3]** : 二つのノード(C1, C2)間の距離 Dist を算出する。単語間距離を決定づける要因がリンク強度と階層内の辺の重みであると仮定し、この二つの要因を考慮したものが以下の式である。

$$\text{Dist}(C1, C2) = \text{IC}(C1) + \text{IC}(C2) - 2 * \text{IC}(\text{LS}(C1, C2))$$

ここで、LS(C1, C2)とは C1 と C2 の情報量の差異を示しておりリンク強度として定義される。

また、この式はあくまで単語間の距離を算出するものであり、類似度を出す式とは目的が対称的である。そのため、WordNet::Similarity 内のメソッドでは上記の式から得られた値を反転させることで類似度として取得している。そのため特例として、IC(C1), IC(C2), IC(LS(C1, C2))が全て0である場合には0を返し、IC(C1)+IC(C2)-2\*IC(LS(C1, C2))=0である場合には0に最も近い距離の逆数を類似度として返すものとする。

**LeacockChodorow (LCH)** : 最短パスの長さを階層の深さで正規化した値を用いる

類似度 Sim を測るための二つの単語の Synset の最短距離 (len(C1, C2)) に依存する

しかし、Is-A リンクと、分類の全体的な深さ Depth による経路長の規模に配慮して制限している

$$\text{Sim}(C1, C2) = -\log(\text{len}(C1, C2) / 2 * \text{Depth})$$

**Lesk (LESK) [4]** : 単語のそれぞれの定義について比較を行う。その際比較に使用する定義は Oxford Advanced Learner's (OAL) から引用していたが、その後 Banerjee と Padersen によって WordNet に適応させた手法が提案された。

WordNet は OAL から品詞の語彙量、意味について改善したものであり多義性についても補完している。Padersen らによる手法では、WordNet で検索された定義文からターゲット単語を含む前後の文脈を対象

とし、その中の単語を順次入れ替えて比較を行う。

今回の実験では Banerjee と Padersen によって提案された WordNet を用いての類似度比較 Lesk を用いるものとする。

**Lin (LIN) [5]** : 二単語に共通する上位概念のうち最下位の概念 LCS (Least Common Subsumer) の選択情報量をそれぞれの Synset が持つ選択情報量で正規化した値を類似度 Sim とする。

二単語間の類似度は情報量の比で表せるものとし、情報量の比を二単語の共通性 (common) の情報量と二つの単語についてそれぞれの定義文 (description) の情報量の比とした式が以下である。

$$\text{Sim}(A, B) = \frac{\log P(\text{common}(A, B))}{\log P(\text{description}(A, B))}$$

この式を WordNet に適応すると、WordNet に登録されている単語 C の存在確率 P(C) を用いて以下のように表現できる。

$$\text{Sim}(C1, C2) = \frac{(2 * \log(P(\text{LCS}(C1, C2))))}{(\log(P(C1)) + \log(P(C2)))}$$

**Path (PATH)** : WordNet 内の IS-A 階層間のノード数を数えることによって得られる。

**Resnik (RES) [6]** : オントロジーとコーパスを結集し、『情報を共有していると予測される』二つの概念間の類似度から導かれる。情報コンテンツとなる WordNet 内の二つの語彙の概念間の類似度 Sim を以下の式によって定義した。

$$\text{Sim}(C1, C2) = -\log(P(\text{LCS}(C1, C2)))$$

**Wu&Palmer (WUP) [7]** : Is-a 階層における LCS の深さ depth, 2つの syneset の深さを考慮している。ここでいう深さとは親ノードからのノード数のことである。

$$\text{Sim}(C1, C2) = \frac{2 * \text{depth}(\text{LCS})}{(\text{depth}(C1) + \text{depth}(C2))}$$

ただしこの算出方法はアプローチを行う構造が IS-A リンクからなる概念階層のみに限られているため、注意が必要であり、単語そのものの意味を比較する際には重みを考慮する必要があるとされている。本研究においては WordNet がもともと概念辞書であるため、上記の式をメソッドとして用いる。

実験で使用される問題は事前に 11 名に解答してもらい、正答率に偏りが無いとされたインターネットのサイト [8] 上のフリーの問題を使用する。なお

WordNet および WordNet::Similarity を使用するにあたって、使用する単語の品詞タグを必要とするため、英語品詞タグ GeniaTagger [9] を用いて事前にデータの品詞タグ付けを行う。ただし、WordNet からの Offset (synset id) 取得を目的とするため、GeniaTagger から得られた分析のうち、品詞タグのみを使用するものとする。

また、実験で使用するデータは問題文全文、設問及び選択肢、重要文の 3 つのデータにそれぞれ品詞タグを付与したものをを用いる。ここでいう重要文とは実験前に人手で抽出した、問題文から解答に必要な情報が含まれている文を指す。

例えば、次のようにデータが与えられている時、

<p>問題文          …We'll Send You A FREE GIFT!          Order a full Wizard Language Program and we'll send you a portable stereo cassette player absolutely free. A great way to learn …</p> <p>設問          What is being offered at no charge?</p>
--

” no charge (料金無し)” とは ” free (無料)” と同意であるので問題文中の “…we'll send you a portable stereo player absolutely free.” を重要文として抽出するものとする。

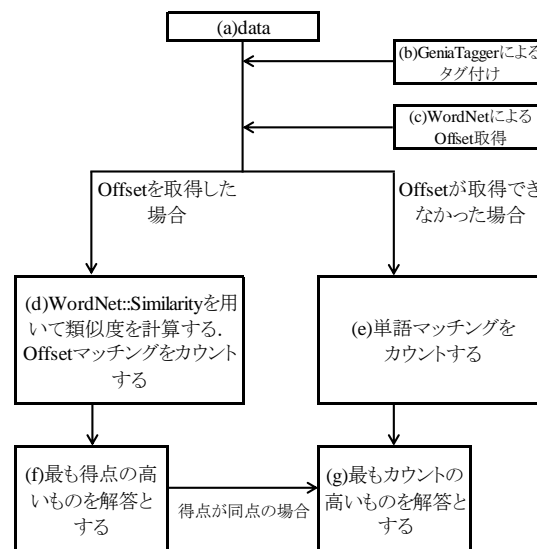


図 1. システムの構成図

以下に検討する解答手法を述べる

(1) 問題文データ，選択肢データからストリングマッチングを用いる手法

- (a) 問題文データ，選択肢データを読み込む
- (e) それぞれの単語についてストリングマッチングを行い，選択肢ごとにカウントを行う
- (g) A~D の選択肢で最も問題文中の単語とマッチした選択肢を解答とする

(2) 重要文データ，選択肢データからストリングマッチングを用いる手法

- (a) 重要文データ，選択肢データを読み込む
- (e) それぞれの単語についてストリングマッチングを行い，選択肢ごとにカウントを行う
- (g) A~D の選択肢で最も問題文中の単語とマッチした選択肢を解答とする

(3) 重要文データ，選択肢データから Offset およびストリングマッチングを用いる手法

- (a) 重要文データ，選択肢データを読み込む
- (b) GeniaTagger を用いて品詞タグを付与する
- (c) WordNet から Offset を取得する
- (d) Offset が取得できた場合 Offset マッチングを行い，選択肢ごとにカウントする
- (e) Offset が取得できなかった場合，ストリングマッチングを行い，選択肢ごとにカウントする
- (f) カウントが最も高い選択肢を解答とする

(4) 重要文データ，選択肢データから単語間類似度とマッチングを用いる手法

- (a) 重要文データ，選択肢データを読み込む
- (b) GeniaTagger を用いて品詞タグを付与する
- (c) WordNet から Offset を取得する
- (d) Offset が取得できた場合単語間類似度を測定しポイントとして選択肢ごとに加算していく。同時に Offset カウントも行い，選択肢ごとにカウントする
- (e) Offset が取得できなかった場合，ストリングマッチングを行い，選択肢ごとにカウントする
- (f) ポイントが最も高い選択肢を解答とする
- (g) (f) でポイントが同率だった場合に，カウントが最も高い選択肢を解答とする。

(5) 問題文データ，選択肢データから単語間類似度と Offset マッチングを用いる手法

- (a) 問題文データと選択肢データを読み込む
- (b) GeniaTagger を用いて品詞タグを付与する
- (c) WordNet から Offset を取得する
- (d) Offset が取得できた場合単語間類似度を測定しポイントとして選択肢ごとに加算して

いく。同時に Offset カウントも行い，選択肢ごとにカウントする

- (e) Offset が取得できなかった場合，ストリングマッチングを行い，選択肢ごとにカウントする
- (f) ポイントが最も高い選択肢を解答とする
- (g) (f) でポイントが同率だった場合に，カウントが最も高い選択肢を解答とする。

3. 実験結果と評価

手法(1)~(5)について実験を行い，正答率を比較する。WordNet::Similarity を用いて解答を行う手法については，各メソッドについてテストデータを解答し，最も正答率が高いものをデータとして用いるものとする。以下はその比較である。

表 1. 各メソッドの正答率比較

メソッド	正解数	正答率 (%)	得点
JCN	4	30.8	5.9
LCH	4	30.8	5.9
LESK	4	30.8	6.3
LIN	2	15.4	5.1
PATH	3	23.1	5
RES	3	23.1	5
WUP	3	23.1	5.4

また，システムの評価方法は，正解数を問題数で割った正答率と，解答候補の順位に対応する得点をつける得点制で評価する。

- 1位：1点
- 2位：0.5点
- 3位：0.3点
- 4位：0.1点

その結果，表 1 に示す値が得られた。

表 2. (1)~(5)によるテストデータの正答率

No	正答数(correct/All)	正答率 (%)	得点
(1)	3/13	23.1	5.4
(2)	3/13	23.1	5.5
(3)	6/13	46.2	7.3
(4)	6/13	46.2	8.1
(5)	4/13	30.8	6.5

(1)~(5)の比較から，問題を解答する際に重要文として判定範囲を狭める手法がより有用であることがわかった。これは TOIEC の問題の特性上，正解以外の設問選択肢が質問とは関係のない文中の記述である場合が多いためだと考えられる。また，メソッドの正答率比較を行った際には単語間類似度を単独で用いたが，No. 4 から類似度が同点の場合にマッチ

ング使用した方が、正答率がよいことがわかる。これは単語間類似度が固有名詞や数詞などに用いることができないためだと考えられる。

次に表2の結果から No. 3, 4, 5 についてテストデータ以外の問題も解答した全体の正答率について比較を行う。

表3. (3)~(5)による全データの正答率

No	正答数(correct/All)	正答率(%)	得点
3	19/45	42.2	24.8
4	25/45	55.6	30.6
5	19/45	42.2	26.6

表3から、(4)の手法において正答率 55.6%という結果が得られた。表2との正答率の大きく異なることから問題数を増やして人の正答率データと比較する必要があると考えられる。

またテストデータにおいて、問題評価のために事前に解答してもらった人間の正答率と比較した。

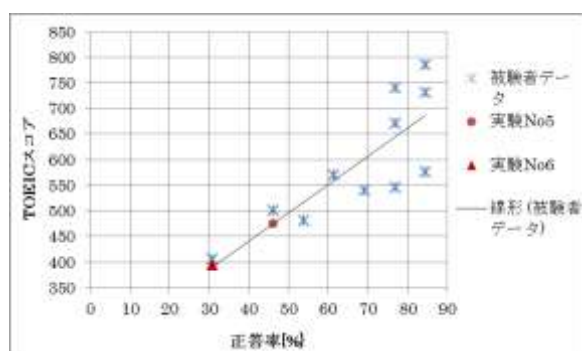


図2. 人間の正答率との比較

図2にて被験者データの線形から実験データの正答率をもちいて手法No4のTOEICスコアを読み取ったところ、およそ475というスコアが得られ、正答率46.2%という値は人間と比べるとまだ十分な意味解析を行えていないと感じた。また、手法No4は人手で判別した重要文を用いているため、完全に自動解答を行った手法No5を比較するとTOEICスコアは400となった。

表1と表2から問題データの量によって同じ解答法でも正答率が大きく異なるため、全データについても人間との正答率比較を行うべきであると考えられる。

#### 4. おわりに

今回の実験において正解の情報を含む重要文の特定が必要であることがわかった。そのため、現在人手で決定している重要文の特定をどのようにシステム化することが今後重要な課題になってくると考えられる。

手法については、今回マッチングと単語間類似度を併用したが、固有名詞や数詞以外の単語の一致度は単語間類似度に大きく依存している。そのため、今後システムの正答率を上げていくなれば、単語間類似度よりもより正確な類似度を測れる手法を検討することが重要だと感じられた。

また、単語間類似度については、本研究で用いたデータでは LESK が最も読解問題を解く上で有効であり、今回比較したメソッドの中で唯一概念・定義文からのアプローチであった。この手法が全ての英語読解問題において有用であるのか特定のサイトのフリー問題だけでなく複数の出典の問題を試し、同時に他のメソッドの情報量、階層構造からのアプローチとの差異を考える必要があると思われる。

#### 参考文献

- [1] WordNet (<http://wordnet.princeton.edu/>)
- [2] WordNet::Similarity (<http://wn-similarity.sourceforge.net/>)
- [3] Jay J. Jiang and David W. Conrath, Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy, In Proc. of ROCLING X, Taiwan, 1997.
- [4] Satanjeev Banerjee and Ted Pedersen, An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet, In Proc. of Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science Volume 2276, pp 136-145, 2002.
- [5] Dekang Lin, An Information-Theoretic Definition of Similarity, In Proc. of Proceedings of the Fifteenth International Conference on Machine Learning, pp 296-304, 1998.
- [6] Philip Resnik, Using Information Content to Evaluate Semantic Similarity in a Taxonomy, In Proc. of IJCAI-95, pp 448-453, Montreal, Canada, 1995.
- [7] Wu and Palmer, VERB SEMANTICS AND LEXICAL SELECTION, In Proc. of the 32nd Annual Meeting of the Associations for Computational Linguistics, pp 133-138, Las Cruces, New Mexico, 1994.
- [8] Webanhvan (<http://webanhvan.com/>)
- [9] GeniaTagger (<http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/>)