

関係の対称性を用いた Relational Search の精度向上法

後藤 友和[†] Nguyen Tuan Duc[†] Danushka Bollegala[†] 石塚 満[†]

東京大学大学院情報理工学系研究科[†]

1 はじめに

近年の検索エンジンの発展は目覚しいが、キーワードマッチングベースでは検索を行うのが困難なタスクも数多く存在する。例えば、アメリカに旅行に出かける際に、「日本でいう富士山のような山を見たいので探したい」と思っても、意味が曖昧すぎて探すのが困難である。Relational Search はこのような問題を解決することができる。Relational Search は、関係 R を持つ単語のペア (A, B) とある単語 C を入力に持ち、 C と R の関係にある単語 D を出力とするタスクである [Kato 09, Duc 10]。例えば、 $\{(ostrich, bird), (lion, ?)\}$ というクエリを入力した場合、結果として「cat」が得られる。 D が cat のとき、それぞれの単語対が「...は大きな～である」という関係 R を持つからである。

以下、Relational Search に用いるクエリ Q_{RS} を、3つの単語 A, B, C を用いて $Q_{RS} = \{(A, B), (C, ?)\}$ と表す。また、最初のクエリから得られた D を用い、クエリ中の他の単語を「？」にしたときに取得が期待できる語のことを予測語という。例えば、 $Q_{RS} = \{(A, B), (C, ?)\}$ から得た D を用いたクエリ $Q_{RS} = \{(A, B), (? , D)\}$ においては予測語 C を得ることが期待できる。

Relational Search は主に Web のデータを用いて実現されているため [Kato 09, Duc 10]、ノイズが混じりやすく、それによる性能低下が起こるといった問題がある。本論文では、Relational Search における予測語の期待性と、関係の対称性による複数の Relational Search のクエリを用いることで検索結果の検証を行い、Relational Search の性能を向上させる手法を提案する。

2 予測語と対称的なクエリによるノイズの影響の削減

$Q_{RS} = \{(A, B), (C, ?)\}$ には対称的なクエリ Q'_{RS} を定義することができる。例えば、 $Q'_{RS} = \{(B, A), (? , C)\}$ は対称的なクエリであり、 Q_{RS} と同様の D が期待できる。 $Q_{RS} = \{(A, B), (C, ?)\}$ において検索結果の単語を D とすると、予測語を得るための対称的な Relational Search のクエリ Q'_{RS} を表 1 に示す sym1 から sym8 までの 8 種類定義できる。ここで、それぞれのクエリを用いたときに取得できる予測語のスコアをそれぞれ s_1 から s_8 とする。また、Relational Search で得られた単語集合に含まれる単語をランキングする際に用いるスコアリング関数を $score(\cdot)$ で表す。ただし $score(\cdot)$ は、適切に単語の正しさを表す必要があり、先行研究では χ^2 値 [Kato 09] や、コサイン類似度 [Duc 10] を用いて $score(\cdot)$ を定義している。本研究では、 (A, B) 間の関係 R と Q_{RS} から得られた D に対し、 R と (C, D) がどれだけ強く結びついているかを表す関数を $score(\cdot)$ として用いる。

D を $Q_{RS} = \{(A, B), (C, ?)\}$ から得た語とするとき、 D が正しい解であれば、 $Q_{RS} = \{(A, B), (? , D)\}$ からは予測語 C を高いスコアと共に得ることが期待できる。逆に、 D が誤った解であれば、 $Q_{RS} = \{(A, B), (? , D)\}$ からは予測語 C を得られ

表 1: 対称的な Relational Search のクエリ

名前	対称的なクエリ Q'_{RS}	予測語	スコア
sym1	$\{(A, B), (? , D)\}$	C	$s_1 = score(C)$
sym2	$\{(B, A), (D, ?)\}$	C	$s_2 = score(C)$
sym3	$\{(C, D), (A, ?)\}$	B	$s_3 = score(B)$
sym4	$\{(C, D), (? , B)\}$	A	$s_4 = score(A)$
sym5	$\{(D, C), (B, ?)\}$	A	$s_5 = score(A)$
sym6	$\{(D, C), (? , A)\}$	B	$s_6 = score(B)$
sym7	$\{(A, B), (C, ?)\}$	D	$s_7 = score(D)$
sym8	$\{(B, A), (? , C)\}$	D	$s_8 = score(D)$

ないか、低いスコアと共に得ることになる。この特性を用いることにより、Relational Search の出力 D の妥当性をチェックすることができる。

予測語は表 1 に示す 6 通りあり、 D を取得するために用いるクエリ 2 種類を加え、計 8 通りのクエリを用いて D の妥当性を評価できる。そして、これら 8 通りのクエリを用いて得られる D の最終的なスコア $FinalScore(D)$ を次のように定義する。

$$FinalScore(D) = \sum_{i=1}^8 u_i s_i \quad (1)$$

ここで、 u_i は、 i 番目のスコアを用いるか否かを表す値であり 0 か 1 の値を取る。なお、本論文では 3 章で実験的に $u_1 \dots u_8$ の組み合わせを求めている。

3 評価実験

Relational Search には評価を行うためのベンチマークが存在しないため、Scholastic Aptitude Test(SAT) データセットを評価に用いた [Bollegala 09]。SAT データセットに含まれる問題の例を表 2 に示す。SAT のアナロジー問題は問題の単語対と、回答の選択肢となる 5 つの候補の単語対から成り、データセット内には全部で 374 問存在する。回答者は問題単語対の關係に最も類似する単語対を候補の単語対の集合から 1 つ選び、回答する。表 2 の場合、ostrich と bird の關係に最も近くなるような候補の単語対は lion と cat であるため、正解は 1.(lion,cat) となる。

Relational Search では、 $Q_{RS} = \{(A, B), (C, ?)\}$ というクエリに含まれる単語対をそれぞれ問題単語対と候補ペアに含まれる単語に当てはめることで SAT の問題を解くことができる。具体的には、 A と B をそれぞれ問題単語対に含まれる

表 2: SAT のアナロジー問題の例

問題単語対	ostrich	bird
1	lion	cat
2	goose	flock
3	ewe	sheep
4	cub	bear
5	primate	monkey

表 3: SAT データセットにおける正解率

番号	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	精度	再現率	F 値	順位
1	0	0	0	0	1	1	1	1	.511	.447	.476	1
2	0	0	0	0	1	0	1	1	.509	.439	.471	2
3	0	0	0	0	1	1	0	1	.506	.433	.467	3
4	0	0	1	1	1	1	1	1	.491	.444	.466	4
5	0	0	1	1	1	0	1	1	.488	.441	.463	5
6	0	0	1	0	1	1	1	1	.488	.439	.462	6
7	0	0	1	0	1	1	1	0	.494	.433	.462	7
8	0	0	1	1	1	1	0	1	.487	.439	.461	8
9	0	0	0	0	1	0	0	1	.511	.420	.461	9
10	0	0	0	0	1	0	1	0	.511	.420	.461	10
11	1	1	1	1	1	1	1	1	.442	.404	.422	56
12	0	0	0	0	1	0	0	0	.509	.382	.437	33
13	0	0	0	0	0	1	0	0	.586	.345	.434	37
14	0	0	1	0	0	0	0	0	.421	.313	.359	234
15	0	0	0	0	0	0	0	1	.420	.305	.353	239
16	1	0	0	0	0	0	0	0	.383	.307	.341	246
17	0	0	0	0	0	0	1	0	.490	.257	.337	250
18	0	1	0	0	0	0	0	0	.336	.262	.294	253
19	0	0	0	1	0	0	0	0	.352	.246	.290	255

単語に置き換え、C を候補ペアの最初の単語に割り当てれば良い。こうして Q_{RS} を作成し、D を検索する。そして、「?」に当てはまった、候補ペアに含まれる単語のうち C でない単語 (ラストワード) のスコアを見る。これを全ての候補に対して行い、それぞれのラストワードのスコアを比較し、最もスコアが高いものを選ぶことによって SAT の正解とする。例えば、表 2 の場合、A = ostrich, B = bird, C = lion とし、D として出てくるラストワード = cat のスコアを見る。これを他の 4 つの候補に対しても行い、取得できたラストワードのスコアを比較する。その結果、最も高いスコアを出したものを SAT の答えとして出力する。システムの出力した SAT の答えと SAT データセットに用意された正解と照らし合わせることで、システムによる結果が正しいかどうかを判断できる。本研究ではこの正解率を見ることで、精度、再現率、F 値を求め、対称性を用いる前後における性能比較に用いた。なお、精度 (precision) は、

$$\text{precision} = \frac{\text{正解数}}{\text{正解が判断できた問題数}} \quad (2)$$

で表され、再現率 (recall) は

$$\text{recall} = \frac{\text{正解数}}{\text{全問題数}} \quad (3)$$

で表され、F 値 (F-measure) は

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

で表される。ここで、精度を表す式にある「正解が判断できた問題数」は、 $Q_{RS} = \{(A, B), (C, ?)\}$ または $Q_{RS} = \{(B, A), (? , C)\}$ において、候補ペア 5 つのうち 1 つでも D としてラストワードを抽出できた問題の数を表す。候補ペア 5 つに渡ってラストワードを抽出できない場合、Relational Search において正解が判断できないため、Relational Search の結果としては適さない。

実験結果を表 3 に示す。番号 1 から 10 は F 値が高い方から 10 個並べたものを、番号 11 は全ての対称性を用いたものを、そして番号 12 から 19 は、それぞれの対称性を 1 つだけ用いたものを表す。組み合わせ (u_1, \dots, u_8) は、式 (1) における u_i に対して、 i 番目のスコア (s_i) を使用する時に $1(u_i = 1)$

を割り当て、使用しない時に $0(u_i = 0)$ を割り当てたものである。例えば番号 1 では、 $u_1 = 0, u_2 = 0, u_3 = 0, u_4 = 0, u_5 = 1, u_6 = 1, u_7 = 1, u_8 = 1$ となり、 s_1 から s_4 の対称性は全く使用せず、 s_5 から s_8 の対称性は全て使用することを表す。これら u_i の組み合わせは $2^8 = 256$ 通りから、全てが 0 の場合を除いた 255 通り存在する。よって F 値の順位は 1 から 255 まで存在する。

本研究では、D に関する対称性を使う前と対称性を使った後での F 値の変化を見ることによって評価を行う。D に関する対称性を使わない場合は表 3 において番号 17 で表される。これは、 $Q_{RS} = \{(A, B), (C, ?)\}$ から得られた D のスコアに等しく、本手法における比較対象となる。また、それぞれの対称性の効果を見るために、対称性を 1 つだけ用いたクエリに関しても正解率を載せている。

4 考察

表 3 を見ると、番号 1 のときに最も F 値が高くなっている。これは、単に全ての対称性を使った場合 (番号 11) よりも良い結果になっている。また、単独で対称的なクエリを用いた場合では、番号 12、つまり $\text{sym6} = Q_{RS} = \{(D, C), (? , A)\}$ を用いたときに最も F 値が高い。また、対称性を使った D の検証を行う前 (番号 17) に比べて F 値は 33.7% から 47.6% へと 13.9% 上昇しており、検証を行う前に比べて 141% の F 値になっていることが分かる。このことから、本手法による D の検証が Relational Search の性能を大きく向上させていることが分かる。

全体の順位で見ると、対称性を用いない場合が 250 位であるため、対称性を用いた方が多くの場合で F 値が向上していると言える。また、組み合わせ係数 u_i のうち、1 になっている係数の数が 1 つだけのものをみると、番号 12 と番号 13 を除く 6 種類が 200 位台後半にあることから、使用する対称性は 1 つではなく複数個使用するのが望ましいことが分かる。また、全ての対称性を用いると最高で 8 倍の時間が必要になるが、全ての対称性を使用するよりも、使用する対称性を選別する方が F 値が上昇していることから、最も F 値の高い番号 1 より、実際には 4 倍程度で済むことが分かる。

また、対称性の選別時に F 値が良かったのは、関連の無い問題単語対と候補単語対に対しても答えを出し易い対称性があったためだと考えられる。SAT の正解候補以外でも予測語が取得できた場合、誤った答えを SAT の正解として出すことに繋がる。

5 おわりに

本論文では、Relational Search の性能を向上させるために、関係の対称性を用いた複数のクエリによるスコアリング手法を提案した。SAT データセットを用いた実験の結果、対称性を使わない場合に比べて F 値が 13.9% 向上した。また、多くの場合、対称性を用いたクエリの方が F 値が向上することが分かった。さらに、全ての対称性を使う場合よりも、半数の対称性に絞った場合の方が F 値が高くなった。このことは、高速・高精度な Relational Search を実現する際に役立つ。

参考文献

- [Bollegala 09] Bollegala, D., Matsuo, Y., and Ishizuka, M.: Measuring the similarity between implicit semantic relations from the web, in *Proc. of WWW'09*, pp. 651–660 (2009)
- [Duc 10] Duc, N. T., Bollegala, D., and Ishizuka, M.: Using Relational Similarity between Word Pairs for Latent Relational Search on the Web, in *Proc. of the IEEE/WIC/ACM Int'l Conf. on Web Intelligence, WI'10*, pp. 196–199 (2010)
- [Kato 09] Kato, M. P., Ohshima, H., Oyama, S., and Tanaka, K.: Query by analogical example: relational search using web search engine indices, in *Proc. of CIKM'09*, pp. 27–36 (2009)