

強化学習を用いたイベント抽出のためのクエリ拡張

谷口元樹 三浦康秀 大熊智子
富士ゼロックス株式会社

{motoki.taniguchi, yasuhide.miura, ohkuma.tomoko}@fujixerox.co.jp

1. はじめに

イベント抽出タスクはテキストから予め定義されたイベントの属性値を抽出することで、イベント情報を構造化する。例えば新聞記事から企業の買収やテロ事件のイベントにおける、買収企業名と被買収企業名や発生場所、犯人の名前、被害者数を抽出する。このように構造化された情報は質問応答や対話などの他の言語処理タスクを解くための知識源として活用されるため、重要である。

イベント抽出には大量の学習データによる教師あり機械学習手法がよく用いられる。しかし、学習データを作成するコストが大きいことやイベントの発生頻度が少ないことから、学習データを大量に作成することが難しく、高い精度を達成できない場合がある。そのような場合には、Webなどの外部情報源から、同一のイベントについて書かれた文書(関連文書)を検索し、関連文書からも属性値を抽出する手法が用いられる。この手法では関連文書を検索するためのクエリの決定方法、関連文書から抽出されたエンティティがイベントの属性値として適切かどうかを判断する方法が課題となる。

この課題に対して、強化学習を適用した研究が行われている[Narasimhan et al., 2016; Sharma et al., 2017]。これらの研究では、検索して得られた関連文書から抽出されたエンティティをイベントの属性値として受け入れるかどうかの決定と、次の文書を検索するためのクエリの決定をエージェントの行動とみなしている。しかし検索するためのクエリは原記事のタイトルを元クエリとしてテンプレートをを用いて拡張されており、追加される単語はイベントによらず固定である。

本論文では疑似適合フィードバックを用いて、イベントに応じたクエリに自動的に拡張するモデルを提案する。疑似適合フィードバックでは元クエリで検索して得られた上位の文書は関連性があると仮定し、それらの文書に含まれる単語を用いてクエリを拡張する。提案モデルにおけるエージェントの行動は、元クエリに対して追加する単語を疑似適合フィードバックで得られた候補単語から選択することである。これにより、イベントに応じた拡張クエリが生成され、関連文書の検索精度が向上することが期待される。さらに提案モデルの方策には行動候補と状態の組合せを考慮したニューラルネットワークを用いている。

本論文の貢献は以下の3点である。

- 疑似適合フィードバックを用いて、イベントに応じたクエリに自動的に拡張するイベント抽出のためのモデルを提案する。
- 従来のテンプレートと比較して、疑似適合フィードバックによる拡張クエリを用いることで関連文書の検索精度が向上することを示す。
- 行動候補と状態の組合せを考慮したニューラルネッ

トワークを用いて方策を学習することで、イベント抽出の精度が向上することを示す。

2. 関連研究

外部情報源を活用した情報抽出

外部情報源から関連文書を検索することで、情報抽出の精度を高める取り組みは盛んに行われている[Wu and Weld, 2010; Ji and Grishman 2011]。しかし、イベント抽出では検索によって得られた文書が目的のイベントの関連記事であるかどうかを判断するイベント共参照解析が必要になる点が他の情報抽出タスクとは異なる。

クエリ拡張

クエリ拡張は検索を対象とした研究が広く行われており、クリックスルーログを用いる手法[Sordoni et al., 2014]、疑似適合フィードバックを用いる手法[Xu and Croft, 1996]などがある。我々の手法では、クリックスルーログなどの他のリソースを必要としないため、疑似適合フィードバックを用いる。

[Nogueira et al., 2017]では疑似適合フィードバックと強化学習を用いて、検索タスクでクエリ拡張を行っており、我々の手法に類似している。しかし、検索とは異なり、イベント抽出では関連文書を検索するだけでなく、目的の属性値が書かれている文書を検索する必要がある点で、さらに難しいタスクになる。また、イベントの属性は複数定義されることが多く、同時に複数の属性値を抽出するため、より複雑なタスクになる点が異なる。

強化学習を用いたイベント抽出

[Narasimhan et al., 2016; Sharma et al., 2017]では外部情報源を活用したイベント抽出タスクをMDP(Markov Decision Process)でモデル化し、強化学習を適用することでメタ学習器を上回る精度を達成した。これらの研究はいずれも関連文書を検索するためのクエリをテンプレートをを用いて拡張しており、イベントによらず固定である。また、クエリの選択には、クエリの言語情報は用いられていない。

3. 提案手法

[Narasimhan et al., 2016; Sharma et al., 2017]と同様に、イベント抽出タスクをMDPでモデル化するが、検索クエリの決定方法が異なる。我々のモデルにおけるエージェントの行動は、クエリテンプレートの選択ではなく、元クエリに追加する単語を候補単語から選択することである。

本章では、まずイベント抽出タスクのMDPによるモデル化について説明する。次に、疑似適合フィードバックを用いた候補単語の作成方法を説明する。最後に、エージェントの方策を学習するニューラルネットワークを説明する。

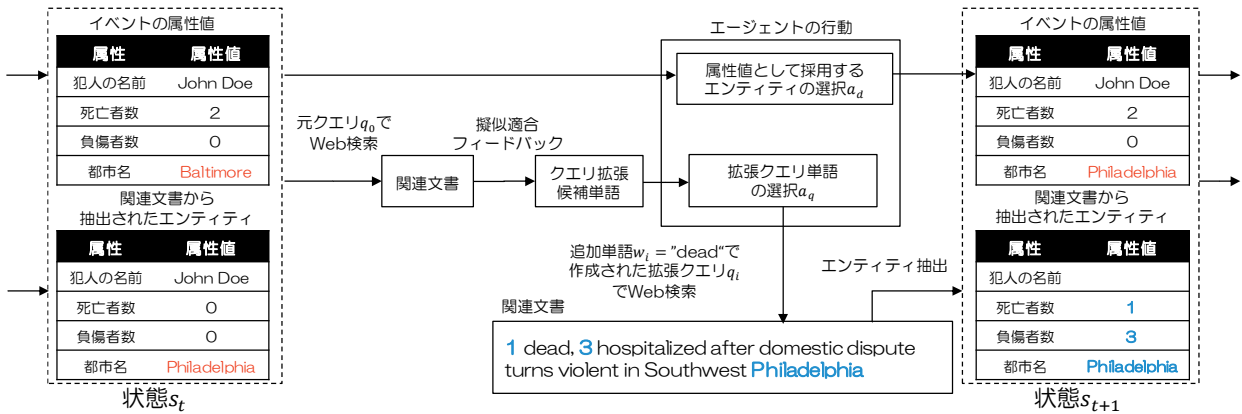


図 1: MDP における状態遷移。赤字はエンティティの選択によって更新される属性値を表す。青字は元クエリ+クエリ追加単語で Web を検索して得られた関連文書から抽出されたエンティティを表す。

3.1 MDP

各エピソードは、はじめに抽出対象となるイベントについて書かれた文書（原文書）が与えられる。その文書から抽出されたエンティティをイベントの属性値として、エージェントの初期状態とする。ステップごとに、エージェントは関連文書から抽出されたエンティティをイベントの属性値として更新するかどうかのエンティティ選択行動 a_d 、次の関連文書を探すためのクエリ拡張単語の選択の行動 a_q を行うことで状態が遷移する。

図 1 にエージェントの状態遷移の例を示す。時刻 t の状態における“都市名”の属性値を Baltimore から、関連文書から抽出されたエンティティである Philadelphia に更新する行動をしている。それと同時に、擬似適合フィードバックによって得られた候補単語から、次の関連文書を検索するための追加単語“dead”を選択する行動をしている。次のステップ t の状態では、更新されたイベントの属性値と、元クエリ+“dead”で検索して得られた文書から抽出されたエンティティに基づく状態から行動を決定する。以降のステップではエピソード終了行動が選択されるまで、同様に逐次的にエンティティの選択と追加単語の選択を行う。

状態: [Narasimhan et al., 2016; Sharma et al., 2017] と同様に、エージェントの状態 s は属性値の信頼度スコア、関連文書から抽出されたエンティティの信頼度スコア、エンティティ周囲の文脈の TFIDF、元文書と関連文書の TFIDF 文書類似度である。

行動: ステップごとに、エージェントは同時に 2 つの行動 $a = \{a_d, a_q\}$ をとる。 a_d は現在の文書から抽出されたエンティティをイベントの属性値として採用するか決定を表し、次の 4 種類の行動候補から選択する。(1) 特定のエンティティを属性値として受け入れる (各属性値あたり 1 つの行動)、(2) 全てのエンティティを採用する、(3) 全てのエンティティを採用しない、(4) エピソードを終了する。 a_q は [Narasimhan et al., 2016; Sharma et al., 2017] とは異なり、次の関連文書を検索するクエリに用いる追加単語の選択である。候補単語 $\{w_1, w_2, \dots, w_{N-1}\}$ から追加単語 w_i を選択し、原記事のタイトルである元クエリ q_0 に対して、追加することで拡張クエリ q_i を生成する行動を a_{q_i} とする。

状態遷移: 決定された行動に基づき、状態が遷移する。エンティティをイベントの属性値として採用するかどうか

かのエンティティ選択行動 a_d から次のステップにおけるイベントの属性値が更新される。同時に、元クエリへの追加単語の選択行動 a_{q_i} に基づき、拡張クエリ q_i で外部情報源を検索し、上位 k 番目の文書を得る。その文書からエンティティを抽出し、次のステップにおける状態とする。ただし、 k は追加単語の選択行動 a_{q_i} がエピソード中に実行された回数を表す。

報酬: [Narasimhan et al., 2016; Sharma et al., 2017] と同様に、現在と一つ前のステップの状態の Accuracy の差を報酬とする。またエピソードの長期化に対するペナルティとして、ステップごとに負の定数項を報酬に加える。

3.2 クエリ拡張の候補単語

クエリ拡張の候補単語は擬似適合フィードバックに基づいて作成する。擬似適合フィードバックでは、元クエリで検索して得られた上位の文書は関連があると仮定する。この仮説に基づき、元クエリを原記事のタイトルとして、上位 M 文書の先頭 K 単語を取得し、クエリ拡張の候補単語とする。したがってクエリ拡張の候補単語数は MK 単語である。追加単語を加えず、元クエリをそのまま検索する場合も加えて、クエリの総数 $N = MK + 1$ となる。

3.3 RLIE-QE

図 2 に提案モデルのニューラルネットワーク RLIE-QE を示す。[Sharma et al. 2016] が提案した RLIE-A3C をベースにしたネットワーク構造を採用するが、イベントに応じたクエリに拡張するために、状態 s と各行動候補 a_{q_i} の組合せからクエリ追加単語の選択に関する方策 $\pi_q(a_q|s)$ を計算する点が異なる。

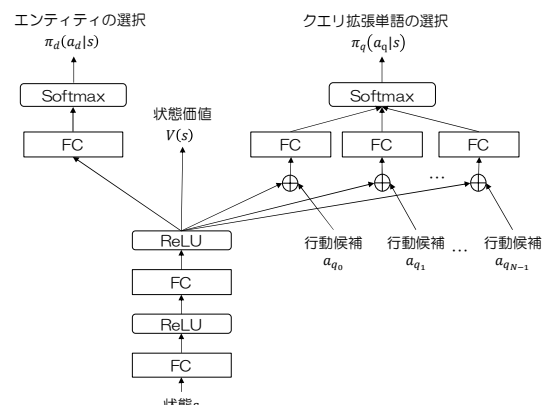


図 2: RLIE-QE のネットワーク構造

			訓練 データ	開発 データ	評価 データ	銃撃犯の 名前	死亡者数	負傷者数	都市名	平均
原文書			372	146	146	0.537	0.697	0.452	0.686	0.593
関連文書	テンプレート クエリ	公開データ N=5	9,829	3,832	4,072	0.560	0.849	0.831	0.863	0.848
		再収集 N=5	22,697	8,874	9,039	0.560	0.872	0.838	0.801	0.837
	擬似適合 FB による 拡張クエリ	M=1, K=4 N=5	20,243	8,877	8,128	0.560	0.895	0.915	0.856	0.889
		M=1, K=10 N=11	45,432	19,505	18,372	0.560	0.919	0.931	0.863	0.904
		M=2, K=10 N=21	90,006	37,256	36,013	0.560	0.919	0.938	0.877	0.911
		M=3, K=10 N=31	134,475	54,966	54,021	0.560	0.930	0.938	0.884	0.917
		M=4, K=10 N=41	178,856	72,554	71,903	0.600	0.930	0.946	0.890	0.922
M=5, K=10 N=51	223,308	90,257	89,844	0.600	0.930	0.946	0.890	0.922		

表 1: コーパス統計量とオラクル Accuracy。“訓練データ”、“開発データ”、“評価データ”の列は各文書数を表す。“銃撃犯の名前”、“死亡者数”、“負傷者数”、“都市名”の列は評価データにおける各属性のオラクル Accuracy を、“平均”は全属性での平均のオラクル Accuracy を表す。

エージェントの状態 s を2層の全結合層に入力し、活性化関数 Rectified linear units (ReLU)を適用することで中間表現 h が得られる。エンティティの選択に関する方策 $\pi_a(a_q|s)$ は中間表現 h に対して、さらに1層の全結合層と Softmax 関数を適用することで得られる。一方、追加単語の選択に関する方策 $\pi_q(a_q|s)$ は中間表現 h と各追加単語 w_i の選択行動 a_{q_i} から計算される。クエリ追加単語の選択行動 a_{q_i} を候補単語 w_i の分散表現で表し、中間表現 h と結合した上で、ユニット数が1の全結合層に入力する。これを各行動 a_{q_i} に対して行い、それらの出力を結合したものに Softmax 関数を適用することでクエリ拡張単語の選択に関する方策 $\pi_q(a_q|s)$ が得られる。ただし、この全結合層のパラメータは全ての行動候補間で共有する。

学習には[Sharma et al. 2017]と同様に Asynchronous Advantage Actor-critic (A3C) [Mnih et al., 2016]を用いる。A3C では複数のエージェントが並列してサンプリングし、方策関数 $\pi(s, a)$ と状態価値関数 $V(s)$ のパラメータを非同期に更新する。これにより、高速かつ安定的に学習を行うことができる。

4. 実験設定

4.1 データセット

[Narasimhan et al., 2016] が公開している Gun Violence archive から作成されたデータセット¹をもとに実験を行う。Gun Violence archive はアメリカにおける銃撃事件のデータベースであり、銃撃事件を報じている新聞記事（原文書）と銃撃犯の名前、死亡者数、負傷者数、事件の発生した都市名の4種類の属性値が記録されている。

学習および評価中に関連文書を検索することは可能であるが、計算効率を考慮して、あらかじめデータを収集した。各原文書に対してタイトルを元クエリとして、擬似適合フィードバックを適用することで候補単語を取得し、取りうる全ての拡張クエリを作成する。この拡張クエリで Bing Search API を用いて Web を検索し、上位 20 文書をダウンロードした。コーパスの統計量は表 1 の左側に示す。また、擬似適合フィードバックの効果を確認するために、従来のテンプレートクエリで学習したベースラインシステムと比較を行う。検索時期や Bing Search API のバージョン更新による影響を確認するため、公開されているデータに加えて、[Narasimhan et al., 2016]が用

いたクエリテンプレートをを用いて再収集を行う。クエリテンプレートは下記の5つであり、<タイトル>には原記事のタイトルを代入して用いる。

1. <タイトル>
2. <タイトル> + (police | identified | arrested | charged)
3. <タイトル> + (killed | shooting | injured | dead | people)
4. <タイトル> + (injured | wounded | victim)
5. <タイトル> + (city | county | area)

4.2 実装

各文書からのエンティティの抽出は[Narasimhan et al., 2016]と同様にロジスティック回帰モデルを用いる。このモデルは単語ごとに4種類のエンティティ（銃撃犯の名前、死亡者数、負傷数、事件の発生した都市名），“その他”の分類を行うことで文書からエンティティを抽出する。同一文書に複数の同種のエンティティが存在した場合は、最も事後確率が高いエンティティをその文書のエンティティとして抽出する。

クエリ拡張候補単語の分散表現は 300 次元であり、Google News を word2vec2 で学習されたモデルを用いる。中間表現 h を計算するための2層の全結合層のユニット数はどちらも 20 に設定する。

A3C は 16 スレッドで学習する。その他のパラメータは [Sharma et al., 2017]と同じものを用いる。

4.3 ベースラインモデル

擬似適合フィードバックによってイベントごとにクエリを拡張した効果を検証するために、ベースラインとなるネットワークとして RLIE-A3C [Sharma et al., 2017]を採用する。RLIE-A3C におけるクエリテンプレートの選択方策 $\pi_q(a_q|s)$ は中間表現 h に対して、一層の全結合層と Softmax 関数を適用することで得られる。実装には公開されているコード¹を用いる。

行動候補 a_{q_i} と状態 s の組合せを考慮したニューラルネットワークの効果を評価するために、テンプレート作成されたクエリで収集したデータを提案手法のネットワークで学習したモデル (RLIE-QE-REDL) も評価する。1つのクエリテンプレートに複数の拡張単語が含まれている場合は、それらの単語の分散表現の平均をクエリ追加単語の選択行動 a_{q_i} とする。

4.4 評価

160 万ステップの学習を行った後に、評価データを用い

¹ https://github.com/adi-sharma/RLIE_A3C

² <https://code.google.com/archive/p/word2vec/>

ネットワーク構造	関連文書データ	クエリ数	銃撃犯の名前	死亡者数	負傷者数	都市名	平均 Accuracy	
RLIE-A3C-ORG	RLIE-A3C [Sharma et al. 2017]	(論文記載)	0.480	0.726	<u>0.742</u>	0.640	0.647	
RLIE-A3C-PUB		公開	0.480	0.733	0.735	0.647	0.649	
RLIE-A3C-REDL		再収集	0.488	0.740	0.717	0.638	0.646	
RLIE-QE-REDL	RLIE-QE (提案手法)	擬似適合 FB による クエリ拡張	N=5	<u>0.520</u>	<u>0.772</u>	0.737	0.642	0.668
RLIE-QE-N5			N=11	0.512	0.753	0.731	0.649	0.661
RLIE-QE-N11			N=31	0.504	0.751	0.726	0.641	0.656
RLIE-QE-N31			N=51	0.480	0.760	0.732	0.641	0.653
RLIE-QE-N51								

表 2: イベント抽出の精度: 下線は各評価指標における最高値であることを示す。“データ”列における“論文記載”は[Sharma et al. 2017]らの論文からの引用であり、“公開”は公開データ、“再収集”はクエリテンプレートをを用いて再収集したデータを関連文書として用いた結果である。

て精度を 50 回繰り返して計算し、上位 5 回分の Accuracy の平均を比較する。

5. 実験結果

提案手法の効果を確認するために、まず擬似適合フィードバックによる拡張クエリによって関連文書の検索精度が向上していることを確認する。その後、提案モデルとベースラインモデルの抽出精度の比較を行う。

5.1 関連文書の検索精度

表 1 の右側に各クエリで検索された関連文書を評価データにおけるオラクル抽出精度を示す。オラクルの抽出精度はエンティティの選択行動 a_d が完全である場合の抽出精度を示す。これは関連文書から抽出されたエンティティに正解となる属性値が含まれるイベントの割合を示している。クエリ数 $N=5$ の場合に、擬似適合フィードバックを用いるとクエリテンプレートをを用いるよりも、オラクルの平均 Accuracy が 5.2pt 高い。これにより、擬似適合フィードバックによるクエリ拡張を用いることで、検索の精度が向上していることがわかる。また、クエリ数 $N=51$ の場合に、さらにオラクルの平均 Accuracy が 3.3pt 向上している。

5.2 属性値の抽出精度

表 2 に各属性値の抽出精度の評価結果を示す。RLIE-A3C は用いているデータによらず、Accuracy に大きな差はなく、検索時期や Bing Search API のバージョンによる影響がないことが確認できる。

RLIE-A3C-REDL と RLIE-QE-REDL を比較すると、平均 Accuracy が 1.0pt 向上している。この結果から、クエリ追加単語の選択候補 a_{q_i} を拡張候補単語 w_i の分散表現で表し、行動候補と状態の組合せを考慮する効果があることがわかる。

RLIE-QE-N5 と RLIE-A3C-REDL を比較してみると、平均 Accuracy が 2.2pt 向上している。属性の中でも、“銃撃犯の名前”は 3.2pt、“死亡者数”は 3.2pt と大幅に向上している。これにより、イベントに応じたクエリで検索することで、関連文書の検索精度を向上するだけでなく、イベント抽出の精度も向上できていることがわかる。

一方で、クエリ数 N が大きくなるにつれて、Accuracy の向上幅は低下している。開発データにおける $\pi(s, a_q)$ を見てみると、クエリ数 N が大きくなるにつれて、単語の選択確率 $\pi(s, a_{q_i})$ の分散は低下しており、どの単語も等しくサンプリングされてしまっていることがわかる。

6. おわりに

本論文ではイベント抽出タスクのための擬似適合フィードバックを用いたクエリ拡張モデルを提案した。提案モデルではイベントに応じた適切なクエリを自動的に生成することができる。実験の結果から、擬似適合フィードバックを用いたクエリ拡張を用いることで、検索の精度が向上していることを示した。また、追加単語が行動候補を表すとして、行動候補と状態の組合せを考慮したネットワークを用いて方策をモデル化することで、平均 Accuracy で 1.0pt 向上することを示した。さらにクエリテンプレートをを用いたベースラインシステムよりも提案手法は平均 Accuracy で 2.2pt 向上することを実証した。

今後の課題としてはクエリ数を増加した場合に、より適切なクエリを選択できるモデルに改良することである。また今回のモデルでは元クエリに追加する単語を候補単語から選択したが、完全に新規のクエリを生成する方法に取り組む。

参考文献

- [Ji and Grishman 2011] Heng Ji and Ralph Grishman. Knowledge base population: Successful approaches and challenges. In NAACL-HLT, pp 1148–1158. 2011.
- [Mnih et al. 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In ICML, pp. 1928–1937. 2016.
- [Narasimhan et al. 2016] Karthik Narasimhan, Adam Yala, and Regina Barzilay. Improving information extraction by acquiring external evidence with reinforcement learning. In EMNLP, pp. 2355–2365. 2016.
- [Sharma et al. 2017] Aditya Sharma, Zarana Parekh, and Partha Talukdar. Speeding up reinforcement learning based information extraction training using asynchronous methods. In EMNLP, pp. 2658–2663. 2017.
- [Sordani et al. 2014] Alessandro Sordani, Yoshua Bengio, and Jian Yun Nie. Learning concept embeddings for query expansion by quantum entropy minimization. In AACL, Vol. 14, pp 1586–1592. 2014.
- [Xu and Croft 1996] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In ACM SIGIR, pp 4–11. 1996.
- [Wu and Weld 2010] Fei Wu and Daniel S Weld. Open information extraction using wikipedia. In ACL, pp 118–127. 2010.