

長い系列データに対する Markov Logic Network の適用

森田 一^{1,a)} 奥村 学^{2,b)} 東中 竜一郎^{3,c)} 松尾 義博^{3,d)}

概要: Twitter 上の対話はチャットのログ等とは異なり、発話にはリプライやリツイートといった明示的な形で他の発話と関連していることを示す情報が含まれる場合がある。そのため、これらの情報を積極的に利用することが Twitter 上の発話間の関係を同定するには重要となる。関係の同定によく用いられている Markov Logic Network (MLN) 上で Twitter 特有の情報を利用し、発話間の関係の有無とその種類の同定を行うモデルを提案する。一方、発話系列には長さに制限がなく非常に長い対話となることもしばしば存在する。このような長い発話に MLN を適用する際に、MLN が大きな問題に対しては計算が困難であることが問題となる。本稿では MLN に対して SVM における動的素性に対応する動的述語を導入することにより、分割した系列に対して近似的に推論が行えるように MLN に対して拡張を行う。

キーワード: Markov Logic Network, Twitter, 対話, 関係同定

Applying Markov Logic Network to Long Sequential Data

HAJIME MORITA^{1,a)} MANABU OKUMURA^{2,b)} RYUICHIRO HIGASHINAKA^{3,c)} YOSHIHIRO MATSUO^{3,d)}

Abstract: We propose a new model for identification of relations between utterances in Twitter conversations. Markov Logic Network (MLN) has been used for such relation identification. There is a large difference between Twitter conversation and other conversation logs. Sometimes, an utterance in Twitter has an explicit information indicating that the utterance have a relation with another utterance, such as reply and retweet. Therefore we employed the MLN for identification of utterance relations by introducing new predicates designed for Twitter. However, the MLN has a problem that cannot scale to large problems when we apply the model for long conversations. We addressed the problem by segmentation of such large scale problems. The segmentation helps to reduce computational complexity of the large scale problems without large loss of accuracy.

Keywords: Markov Logic Network, Twitter, conversation, relation identification

1. はじめに

近年, Twitter に代表される micro blog 上でコミュニケーションが行われることが増えてきており, 容易に大量の対

話データを収集することが可能となった。Twitter 上でのユーザ間の対話には, その時々ニュースやイベントに関する議論や意見を異にするユーザ同士の議論が含まれている。大量の対話データの収集により, これらの対話に含まれる議論の収集もまた可能となっている。一方で, 収集した膨大な議論からユーザが興味を持つ議論を提示する, またはある議論に関連する別の議論を提示するため, 議論の可視化が求められている。

すでに広く一般に利用されている Twitter 上の議論の再利用を可能にするツールとしては, Twitter 上のツイートをまとめるサービスである Togetter [1] が存在する。Togetter 上では, ニュースやイベントに対するユーザの反応, ユーザ

¹ 東京工業大学
Tokyo Institute of Technologies
² 東京工業大学 精密工学研究所
Precision and Intelligence Laboratory, Tokyo Institute of Technologies
³ NTTメディアインテリジェンス研究所
Media Intelligence Laboratories
a) morita@lr.pi.titech.ac.jp
b) oku@pi.titech.ac.jp
c) higashinaka.ryuichiro@lab.ntt.co.jp
d) matsuo.yoshihiro@lab.ntt.co.jp

間の議論等の様々なツイートがユーザ自身によってまとめられている。これは議論の再利用や可視化に対する潜在的な需要を示している一方で、あくまで人手によるまとめであるため、網羅性や中立性には問題が残っている。

議論の収集及び可視化を行うことで、議論内容の再利用や、一覧などを行いやすくすることができる。しかし議論を可視化するには、議論の中で各発言がどのような役割を果たしているかを判別する必要があり、各発言間の関係を同定することが必要となってくる。そこで、関係の同定を行うタスクに多く利用されている Markov Logic Network (MLN) [8] を用いて、Twitter 上の発言間の関係同定を行う。

Markov Logic Network は Semantic Role Labeling[9], 述語項構造解析 [12] および質問応答サイト中の発言間の関係同定 [10] など、多くの要素間の関係を同定するタスクに用いられてきた。これらのタスクと今回の Twitter 上の発言間の関係同定で異なる点は、

- (1) Twitter に特有の情報により、発言間の関係が表される点。
 - (2) 非常に長い対話を入力として扱う場合がある点。
- の二点である。Twitter では一般的な IRC のログや音声対話の記録などと異なり、一部の発言については明示的に言及先の発言を示す情報が与えられるため、これらの情報を積極的に利用することが Twitter 上の発言間の関係を同定する際には重要となる。

また、多数のユーザによる対話であっても、そのすべてのユーザ間で応答があるような全体が密接に関わっている状況は稀である。MLN では扱う問題が大きくなるほど計算が困難になるが、このように発言間の関係は対話のサイズが大きくなるほど疎になるため、対話を適切に分割することによって問題を分割することができる。

この二点の問題点に対し、本稿では Twitter に特有の情報を利用した MLN による発言間の関係同定モデル、および、長い発言系列に対して分割を行い、近似的に MLN を適用する手法を提案する。

2. 関連研究

対話中の発言間に対する関係同定についてはこれまで多くの研究がなされている [4], [5]。今までで手に入れることが比較的容易だった議事録中の対話に対しては、対話間の賛否について同定を行う研究が多く、特に、Morita らは構造学習 [2] を用いて関係同士の依存関係を考慮した学習を行う同定モデルを議事録中の賛否関係について提案しているが [6]、大域的な関係については一定の窓幅内でしか扱っておらず、提案手法に比べると局所的な推論にとどまっている。

また、神保らは質問応答サイトのための関係同定モデルを提案している [11]。彼らのモデルは Support Vector Machine (SVM) を用いて関係を同定しているため、推論

は局所的である。一方、Yokonon らは本稿と同じく Markov Logic Network (MLN) を用いて質問応答サイトの発言間の関係同定が行なっている。MLN を用いることにより、関係間に推移律のような制約を導入し大域的な推論を行うことで、回答間の論理的な関係を同定することを目的としている。提案手法では、Yokonon らが回答間について論理関係を定義したことに対し [10]、発言間について関係を定義し、この関係の種類と関係自体の有無を同定の対象としている。

MLN を用いて要素間の関係に対する同定を行う研究は多く、Poon and Domingos は共参照解析に MLN を用いており [7]、吉川らは述語構造解析に対して MLN を用いている [12]。

3. Markov Logic Network

Markov Logic Network (MLN) とは Richardson らによって提案された、一階述語論理と Markov Network を組み合わせたものである [8]。一階述語論理ではドメインに対する知識を論理式として表現する事ができ、述語の集合である可能世界 y が成立するかどうかを規則として与えられた論理式が充足可能かどうかで表すことができる。一階述語論理と Markov Network を組み合わせることにより、確実に成り立つ論理的な関係だけではなく、確率的に表される知識に対しても述語論理による記述を行うことができるようになる。

MLN では、確率的な知識を表すために、その知識を表す論理式 ϕ に対して重み w を割り当て、不確実な知識を用いた推論を表現する事ができる。与えられた定数 $C = \{n_1, n_2, \dots\}$ と MLN が含む重み付き論理式の集合 $M = \{(\phi_1, w_1), (\phi_2, w_2), \dots\}$ から、次式のように可能世界 y に対する確率分布を与える、

$$P(y) = \frac{1}{Z} \exp \left(\sum_{(\phi, w) \in M} w \sum_{c \in C^{n_\phi}} f_c^\phi(y) \right). \quad (1)$$

ここで、可能世界 y とは定義された述語に定数を代入したある基底述語の集合を表し、 C^{n_ϕ} は述語が含む引数の個数を、 Z は正規化項を表す。例を挙げて説明すると、定数は各発言の $\text{id } u_1, u_2$ を表し、述語 $\text{precede}(u_1, u_2)$ は u_1 が u_2 より以前に発言された、述語 $\text{hasrelation}(u_1, u_2)$ は発言間に応答関係がある、というように発言についての何らかの情報を表す。論理式 ϕ は $\text{precede}(u_1, u_2) \Rightarrow \text{hasrelation}(u_1, u_2)$ というように述語間の関係を表す。述語は観測述語 (observed predicates) と潜在述語 (hidden predicates) に分けることができ、観測述語とは与えられたデータから得られる述語の集合で、潜在述語は一般的な分類問題におけるラベルに対応し、データから推論によって得たい述語の集合である。MLN は観測述語から潜在述語の推論を以下のように行う：

$$\hat{y} = \operatorname{argmax}_{y \in \mathcal{Y}} p(y|x) = \operatorname{argmax}_{y \in \mathcal{Y}} s(y, x)$$

$$s(y, x) = \sum_{(\phi, w) \in M} w \sum_{c \in C^{n_\phi}} f_c^\phi(y, x). \quad (2)$$

対数線形モデルとして見た場合、各可能世界 (y) に対して、(y, x) に含まれる論理式の数 $s(y, x)$ が素性となった、構造推定の問題として捉えることができる。

本稿で実装として用いた markov thebeast では、この事後確率の最大化を Integer Linear Programming (ILP) の問題として解き、対数線形モデルの重みを 1-best Margin Infused Relaxing Algorithm (MIRA)[3] を用いて学習している [9]。最悪の場合には、定数 C に対して $|O(C^\alpha)|$ の Markov Network 上のノードが存在し、ILP として表現する際に各ノードに対して中間変数を用意する必要があるため、MLN が扱う問題のサイズに対して急速に問題を解くことが困難になってしまうという問題が存在する。Riedel らは Cutting Plane Inference (CPI) を用いることにより、推定に必要な時間を大幅に短縮している [9] が、CPI では、最適解に関与しない論理式を扱いを大きく省略することができるが、イテレーションが収束するまでには、最適解に関与するすべての論理式について計算を行う必要があるため、最適解に含まれる論理式の数自体が膨大な場合には、大きな計算量及び計算空間が必要となる。このため、本稿では問題そのものを分割することにより、大きな問題に対して MLN が適用できなくなる問題を近似的に解決する。

4. 発話間の関係

本研究では同定すべき発話間の関係を、発話間の関係の有無とその関係の種類として、5 種類の発話間の関係を定義する。まず発話間に関係があるとは、ある発話 2 がそれ以前の発話 1 を受けて発話されている関係を指し、発話 1 に対する直接的な応答である場合と、発話 1 に対する直接的な応答でなくとも、その内容を踏まえた推論や経緯にそった発言である場合が含まれる。この関係は Twitter 中で明示的に関係が表されるリプライやリツイート等に限らず、同ユーザ間の発話でも内容を踏まえている場合には関係があるとする。直接的な応答がある場合とは、例えば次のような場合を表す：

発話 1: ユーザ 1 タブレットはあまり大き過ぎないほうが持ちやすいし軽くて使いやすい。

発話 2: ユーザ 2 @ユーザ 1 重いと長時間持った時に腕が痛くなるのが嫌ですね。

Twitter 上では特定のユーザに対して “@ユーザ名” として、発話内で相手ユーザの名前を書くことで対象のユーザに対する発言であることを明示することができる。この場合はユーザ 2 が明示的に発話 1 に対して直接応答を行っており、関係があると判断する。一方で、明示的な関係が無く応答と判断される例としては、

発話 1: ユーザ 1 中途半端なサイズの端末は何が嬉しいのかわからない。

発話 2: ユーザ 2 中途半端という人もいるけど、7 インチくらいのサイズが読書にはちょうどいいと思う。

発話 3: ユーザ 2 それ以上大きいと重いし。

発話 1 と発話 2、発話 2 と発話 3 がこの場合関係があると判断する。これは、発話 2 はユーザ 1 に対する発話ではないが、発話 1 で提示された疑問について発話 2 が答えており発話内容を踏まえていると判断できるためである。また、発話 3 は自身の一つ前の発話に対して継続する形でなされており、このような場合も発話間に関係があるとする。

次に、発話間の関係の種類について説明する。本稿では先に述べたように発話間に 5 種類の間関係を定義する。表 1 に発話間の関係とその説明を示す。発話間に関係が有るとした場合は、このいずれかの関係を必ず持つ場合に限られる。

表 1 発話間の関係の種類

| 述語 | 説明 |
|-----|---|
| 矛盾 | 前の発話と異なる内容を述べており、意見や情報に不一致がある場合。 |
| 類似 | 前の発話に同意、または同様の内容を述べている場合。 |
| 演繹 | 前の発話の内容を元として、推論による内容の追加あるいは新しい情報の追加がある場合。 |
| 帰納 | 前の発話の内容をまとめた内容を述べている場合。 |
| その他 | 上記のいずれにも当てはまらない場合。 |

5. Twitter 上の発話間の関係同定

提案モデルでは発話間の関係同定として、発話間の関係の有無と発話間の関係の種類を推定を行う。この 2 つの推定対象について、それぞれ表 2 のように対応する潜在述語を用意する。hasrelation(i, j) では発話の組に対して関係の有無か無いかの二値分類、relation(i, j, t) では発話の組に対してその他を含むどの関係に当たるのかを推定する多値分類を行う。

表 2 潜在述語

| 述語 | 説明 |
|-----------------------|-----------------------------|
| hasrelation(i, j) | 発話 i と発話 j に関係がある。 |
| relation(i, j, t) | 発話 i と発話 j が関係 t をもつ。 |

潜在述語の推論のために用いる観測述語として、Yokono らの提案している観測述語を元に、Twitter 特有の情報を加えた述語の集合を用いた。観測述語として用いた述語の例を表 3 に示す。類似度や間隔は一定の範囲で区切り、範囲の種類自体も定数として用意している。

これらの観測述語を用いて、潜在述語を推論するための重み付き論理式を提案する。引数の前についている “+” は、その引数ごとに論理式に付けられる重みが異なること

表 3 観測述語の例

| 述語 | 説明 |
|-------------------|---|
| tweet(i) | 発言 i がツイートである. |
| first(i) | 発言 i が系列の内最初に発言された. |
| unigram(i,u) | 発言 i が unigram u を含む. |
| bigram(i,b) | 発言 i が bigram b を含む. |
| antonym(i,j) | 発言 i と発言 j が互いに反意となる語を含む. |
| timegap(i,j,s) | 発言 i と発言 j 間の間隔が s である. |
| similarity(i,j,s) | 発言 i と発言 j 間の類似度が s である. |
| inreplyto(i,j) | 発言 i の in_reply_to タグで発言 j が指定されている. |
| reply(i,j) | 発言 i が発言 j のユーザ名を発言内に含む. |
| retweet(i,j) | 発言 i が発言 j に対するリツイートとなっている. |
| visible(i,j) | 発言 i が発言 j より先に発言され、発言 i を見ることができる状態であった. |

を示す. ここで, inreplyto を確実に真となる重み無しの論理式ではなく, 重み付き論理式としているのは, in_reply_to タグがつけられていたとしても, 必ずしもそのツイートに対する応答ではなく, 発言内容はタグが示すより前のツイートに対する応答である場合や, 発言内容からみて応答とは判断されない場合が多く存在するためである.

$$\begin{aligned} \text{inreplyto}(i, j) &\Rightarrow \text{hasrelation}(i, j) \\ \text{precede}(i, j) \wedge \text{timegap}(i, j, +s) &\Rightarrow \text{hasrelation}(i, j) \\ \text{precede}(i, j) \wedge \text{unigram}(i, +a) \wedge \text{unigram}(j, +b) &\Rightarrow \text{relation}(i, j, +t) \end{aligned}$$

また, 重み無しの論理式として, 必ず真となる述語間の関係を表す. ここでは用いた論理式の例を以下に示す.

$$\begin{aligned} \text{hasrelation}(i, j) &\Rightarrow \exists t(\text{relation}(i, j, t)) \\ \text{hasrelation}(i, j) &\Rightarrow \forall k(k \neq i \Rightarrow \neg \text{hasrelation}(k, j)) \\ \text{precede}(i, j) \wedge \forall s(\neg \text{timegap}(i, j, s)) &\Rightarrow \neg \text{hasrelation}(i, j) \end{aligned}$$

6. 発言系列の分割

ここでは, 発言系列が長く推論を行うことが困難になる場合に必要となる, 発言系列を分割して推論を行うための手法について説明する. 一般的に, SVM などの二値分類器を系列に対して適用する際には, 系列のある時点に対する分類結果を次の時点で素性として用いる動的素性を用いる. この動的素性と同様の考えを, 系列を一定数に分割した MLN に対しても用いる.

本稿では分割した系列に対して順に推論を行い, 推論結果を次の分割された系列で述語として用いることにより, 長い系列に対して近似的に推論を行う. 動的素性に対応する, 以前の系列に対して推論した結果を表す述語を表すため, 新たに動的述語を MLN に導入する. ここで, N 分割された系列を $S = \{s_1, s_2, \dots, s_i, \dots, s_N\}$ とする. 動的述語とは, s_i では潜在述語 p_h , s_{i+1} では観測述語 p_o として振る舞う述語として記述され, s_i で潜在述語 p_h として推論さ

れた真偽が s_{i+1} では観測述語 p_o として追加される.

この動的述語を利用することで, 可能世界をまたいで推論結果を参照することができるようになる. 単純に系列を分割した場合, 異なる分割間の述語に対する論理式を推論に用いることができなくなってしまうが, 動的述語により系列間で推論結果を共有し, またいだ分割の両方である述語に対する推論を行うことで分割をまたぐ論理式も推論に用いることができる. このため, 分割された系列に適切に推論を行うには系列が重複した区間を持つように分割を行う必要がある. ここでは, 長さ M である系列の分割数 N を 3 とし, 各分割間で半分づつの発言が重複するように図 1 のように系列の分割を行った. 分割間で半分づつ発言が重複するように分けると, 各分割が含む発言の範囲は, 発言 1 から発言 $\frac{2M}{4}$, 発言 $\frac{M}{4} + 1$ から発言 $\frac{3M}{4}$, 発言 $\frac{2M}{4} + 1$ から発言 M となる. このうち, 分割間で重複しない発言と, 重複する発言の二度目を最終的な発言に対する推論結果とし, 一度目の重複する発言に対する推論を仮推論と呼ぶ. 仮推論で推論された動的述語を二度目の推論で観測結果として用いることで, ここではある区間の発言で分割をまたがる範囲の論理式を考慮した推論を行う.

SVM に動的素性を加える場合とは異なり, MLN ではさらに述語に対応する論理式を追加することで, 初めて推論に動的な推論結果を利用することが可能になる. このため以下のような動的述語を推論に用いた. まず重み無しの論理式として下の 2 つの論理式を用いる:

$$\begin{aligned} \text{hasrelation}(i, j) &\Rightarrow \text{hasrelation}_h(i, j) \\ \text{relation}(i, j, t) &\Rightarrow \text{relation}_h(i, j, t) \end{aligned}$$

この 2 つの論理式によって, ある分割での推論結果を動的述語に移す. hasrelation_h と relation_h が真になった時, 次の分割では hasrelation_o と relation_o が観測述語に追加される. この観測述語を次の分割での推論に役立てるため, 次の重み付き論理式を用いる:

$$\begin{aligned} \text{hasrelation}_o(i, j) &\Rightarrow \text{hasrelation}(i, j) \\ \text{relation}_o(i, j, +t) &\Rightarrow \text{relation}(i, j, +t) \end{aligned}$$

7. 実験設定

提案手法の訓練および評価を行うための Twitter 上の対話データを以下のように収集した. Twitter 上の議論を効率的に収集するため, 本稿では together に集められているまとめのうち, Twitter で行われた議論がまとめられているものを収集する. 各まとめに含まれているツイートを Twitter から再度取得しなおし, ツイートの in_reply_to を含む属性, 各ツイートをリツイートしたユーザの一覧, 各ユーザがフォローしているユーザの一覧をそれぞれ収集しなおした. 時間の経過によりまとめに含まれているツイートが削除される場合やフォロー・フォローワの関係が変

図 1 系列の分割方法

| 発話の範囲 | $1 \dots \frac{M}{4}$ | $\frac{M}{4} + 1 \dots \frac{2M}{4}$ | $\frac{2M}{4} + 1 \dots \frac{3M}{4}$ | $\frac{3M}{4} + 1 \dots M$ |
|-------|-----------------------|--------------------------------------|---------------------------------------|----------------------------|
| 分割 1 | 推論 | 仮推論 | - | - |
| 分割 2 | - | 推論 | 仮推論 | - |
| 分割 3 | - | - | 推論 | 推論 |

表 4 実験結果

| | 関係の有無 | | | 関係の種類 | | | 実行時間 |
|------|-----------|--------|-------|-----------|--------|-------|---------|
| | Precision | Recall | F1 | Precision | Recall | F1 | ILP |
| 分割前 | 0.525 | 0.634 | 0.574 | 0.152 | 0.090 | 0.113 | 11064ms |
| 分割後 | 0.673 | 0.431 | 0.529 | 0.180 | 0.116 | 0.141 | 2711ms |
| 分割後* | 0.617 | 0.424 | 0.503 | 0.160 | 0.110 | 0.131 | - |

化した場合も存在するが、現在手に入る情報に従って対話データを作成した。収集された対話データに対し、4.1 節で述べた基準にしたがい関係のタグ付けを行い実験で用いるデータセットとしている。収集された対話データは全部で 12 件あり、発話数は 21 発話から 187 発話であった。この対話データのうち 8 件を訓練用、4 件を評価用に用いた。このうち、訓練用データに含まれた発話数の最も大きな 187 発話の対話データは分割を行わない場合には学習を実行することができなかつたため、比較のための実験データからは外している。

この対話データを用いて、分割を行った場合と行わない場合それぞれの訓練及び評価を行った際の精度及び速度を比較する。以下の各実験は CPU:Xeon X5675 x2, メモリ:96G の Ubuntu サーバ上で行った。

8. 結果

実験の結果を表 4 に示す。分割により範囲が重複している箇所のうち、一度目に推論を行った仮推論の結果を“分割後*”に示した。この結果を動的述語として用いた二度目の推論結果が“分割後”に示されている。関係の種類を当てる問題では F1 がやや下がったものの、関係の種類を当てる問題では逆にすべての指標で性能が向上しており、推論全体としては、分割を行ったあとも分割前とあまり変わらない精度で関係を同定することができた。また、“分割後*”の結果を見ると、分割を行っただけで動的素性を推論に用いていないにも関わらず、関係の種類を当てる性能が向上している。これは、元々分割前の時点で関係の種類を当てる問題は事例が少ないこともあり、あまりうまく学習が行えておらず性能が低かったことと、分割を行ったことにより関係の種類についての事例が重複して学習されることにより、元々数の多かった関係の有無の同定に比べ、やや関係の種類の同定を重視した学習が行われたためではないかと考えている。次に、“分割後”と“分割後*”の結果を比較すると、一度目の推論より動的述語を利用して再度推論を行った際の結果が全ての指標をにおいて上回っている。この結果から、動的述語を導入することにより、仮推論の時点

ではうまく出来ていなかった推論を補強できていることがわかる。実行時間をみると、整数計画問題を解くために要した時間は分割により大きく削減されている。訓練時は分割を行わない場合と比べ余分に時間がかかったものの、分割によりモデルの更新が行われる回数が増えることによるオーバーヘッドと考えられる。

9. おわりに

本研究では MLN を用いた Twitter 上の発話間の関係にたいする同定モデルを提案した。また、長い発話系列に対して MLN による推論を行うことができないという問題に対し、動的述語を MLN に導入し、系列を分割することにより高速かつ精度よく推論が行えることを実験により示した。今後はより詳細な述語および論理式の検討と、同定した関係に基づく議論の可視化の検討を行いたいと考えている。

参考文献

- [1] Togetter. <http://togetter.com/>.
- [2] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, December 2006.
- [3] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991, March 2003.
- [4] Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. Identifying agreement and disagreement in conversational speech: use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.
- [5] Dustin Hillard, Mari Ostendorf, and Elizabeth Shriberg. Detection of agreement vs. disagreement in meetings: training with unlabeled data. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003-short papers - Volume 2*, NAACL-Short '03, pages 34–36, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.
- [6] Hajime Morita, Hiroya Takamura, and Manabu Okumura. Structured output learning with polynomial

- kernel. In *Proceedings of the International Conference RANLP-2009*, pages 281–286, Borovets, Bulgaria, September 2009. Association for Computational Linguistics.
- [7] Hoifung Poon and Pedro Domingos. Joint unsupervised coreference resolution with markov logic. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08*, pages 650–659, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
- [8] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, February 2006.
- [9] Sebastian Riedel. Improving the accuracy and efficiency of map inference for markov logic. In *Proceedings of the 24th Annual Conference on Uncertainty in AI (UAI '08)*, pages 468–475, 2008.
- [10] Hikaru Yokono, Takaaki Hasegawa, Genichiro Kikui, and Manabu Okumura. Identification of relations between answers with global constraints for community-based question answering services. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 920–928, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.
- [11] 神保 一樹, 高村 大也, and 奥村 学. 質問応答サイトにおける発言間の関係同定. In *人工知能学会第 24 回全国大会 (JSAI2010)*, pages 3D3–3, 2010.
- [12] 吉川 克正, 浅原 正幸, and 松本 裕治. Markov logic による日本語述語項構造解析. 情報処理学会研究報告. 自然言語処理研究会報告, 2010(5):1–7, 2010.