フレーズ知識補完と生成の同時学習

斉藤いつみ　西田京介　浅野久子　富田準二
日本電信電話株式会社 NTT メディアインテリジェンス研究所
{saito.itsumi, nishida.kyosuke, asano.hisako, tomita.junji}@lab.ntt.co.jp

1 はじめに

本研究では，フレーズ間の知識補完問題に取り組む．一般に知識補完問題とは，あらかじめ与えられた知識グラフをモデル化し，知識グラフ上にリンクが存在しないノード同士の関係性を判定するもので，新たな知識の獲得などに用いることができる．知識グラフ中の情報は $\langle t_1, r, t_2 \rangle$ のように三つ組で表され，$\langle cat, hasA, eye \rangle$ などといった知識が存在する．t_1,t_2 がノード，r がノード間の関係を表し，ノードにはエンティティやテキストが与えられる．ここで，r(=hasA) が欠落している時に，cat と eye の 2 つからどのような関係があるかを予測することが知識補完の一例である．知識グラフの情報は，他の応用タスクにおいても外部知識として用いられるようになってきている．例えば対話システムや質問応答などの応用タスクにおいてこのような外部知識を用いて頑健な応答を生成する研究も増えてきており [9, 6]，知識グラフのカバー率を上げていくことは言語の深い理解において重要な問題であると考えられる．

これまでの知識補完問題においては，WordNet や FreeBase などを用いて，ノードが全て既知の閉じたグラフに対する知識補完のモデルが多く検討されてきた [2, 8]．しかし，ConceptNet などのノードがフレーズの知識グラフを想定する場合，事前に全てのノードを網羅することは難しく，未知のノードに対しても頑健に関係性を判定できるモデルが必要となる．Shi ら [7] は，このような未知のノードを含む知識補完問題を次のように定義している．

定義 1 (知識補完問題) 欠損を含む知識グラフ $\mathcal{G} = (E, R, T)$ が与えられたとき，与えられたグラフ $\mathcal{G}$ に存在しない三つ組 $T' = \{\langle t_1, r, t_2 \rangle | \langle t_1, r, t_2 \rangle \notin T, t_1 \in E_i, t_2 \in E_i, r \in R\}$ を見つけ補完する．ここで，E はフレーズの集合，R は関係ラベルの集合，T は $\mathcal{G}$ 中 に存在する三つ組の集合，E は E_i の部分集合である．

ここで，閉じたグラフ上の知識補完問題とは，上記の定義において $E = E_i$ とする場合の問題設定を指す．Shi ら [7] も指摘しているように，現実の問題においては未知ノードを含む問題設定が多いと考えられる．Li ら [5] は，未知ノードを含む問題設定として，ノードがフレーズの知識補完問題について，三つ組 $\langle t_1, r, t_2 \rangle$ が与えられたときその組み合わせが成り立つか否かの 2 値判別問題として知識補完問題を具体化した．Li ら [5] の手法はシンプルなニューラルネットモデルで 2 値分類の問題において高い精度を示しているが，フレーズのベクトル表現の改良や外部の知識の利用などを行うことでさらなる精度向上が期待できる．

本研究では，Li ら [5] と同様に三つ組の 2 値判別問題としてフレーズ知識補完問題に取り組むが，フレーズ知識の生成問題という新しいタスクを導入し同時に推定する新しい手法について提案する．本研究の貢献を下記に示す．

- フレーズ知識の生成問題という新しい問題を定義し，フレーズ補完問題と同時に解く手法について提案した．
- 三つ組知識について，人手アノテーションデータだけでなく非アノテーションデータを用いる方法について提案した．
- 実験により，知識補完問題において提案手法が state-of-the-art の精度を達成した．

2 タスク設定

本研究では，フレーズ知識補完とフレーズ知識生成という二つのタスクを扱う．それぞれのタスク設定について，説明する．

問題 1 (フレーズ知識補完問題:識別) 三つ組 $\langle t_1, r, t_2 \rangle$ の入力に対し，その組み合わせが成り立つか否かの 2 値分類結果を出力する

問題 2 (フレーズ知識補完問題:ランキング) フレーズ $t_1 \in E_i (t_2 \in E_i)$ と関係ラベル $r \in R$ が与えられたとき，$t_1(t_2)$, r との関係性スコアで $t_2(t_1)$ をランキングする．

問題 3 (フレーズ知識生成問題) フレーズ $t_1(t_2)$ と関係ラベル $r \in R$ が与えられたとき，$t_1(t_2)$ と r の関係が成り立つ任意長のフレーズ $t_2(t_1)$ を生成する．

問題 2 については，問題 1 のモデルを使って実現することも可能であるため，今回は従来研究 [5] と同様に問題 1 に取り組む．問題 3 については我々が新規に設定したタスクである．

3 提案手法

提案手法の全体像を図 1 に示す．既存手法と提案手法の大きな違いは，提案手法が知識補完モデルだけでなく知識生成モデルを同時に推定する点である．具体的には，フレーズをベクトル化するための LSTM と語彙，関係ラベルの embedding を知識補完モデルと知識生成モデルで共有する．それぞれの具体的な構成を下記に示す．

3.1 フレーズ知識補完モデル

モデルの基本的な構造は Li ら [5] と類似したモデルを使用するが，入力となるベクトルの生成方法が異なっている．以下具体的に説明するため，まず Li らの提案したモデルについて説明する．

Li ら [5] は，任意の三つ組 $\langle t_1, r, t_2 \rangle$ が与えられた時，三つ組の信頼度スコア $\mathrm{score}(t_1, r, t_2)$ を推定するモデルをニューラルネットワークを用いて次のように定義している．

$$\mathrm{score}(t_1, r, t_2) = W_2 g(W_1 v_{in} + b_1) + b_2 \quad (1)$$

ここで，$v_{in} = \mathrm{concat}(v_{12}, v_r)$，$v_{12} \in \mathbb{R}^d$ は t_1,t_2 を結合した単語列のベクトル表現，$v_r \in \mathbb{R}^d$ は関係ラベル r のベクトル表現を表す．g は非線形関数を表し，本研究では ReLU を用いる．最終層のスコアは 1 次元の出

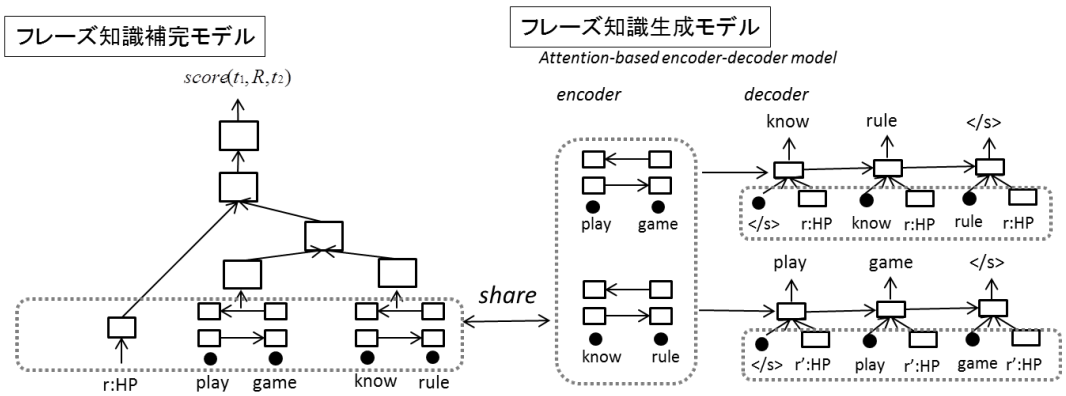

図 1: 提案手法のモデル構造．提案モデルは，$\langle t_1 =$ “play game”$, r =$ “HasPrerequisit”$, t_2 =$ “know rule”$\rangle$ のスコアを出力する知識補完モデルと，$\langle t_1, r\rangle$ から t_2, $\langle t_2, r'\rangle$ から t_1 をそれぞれ生成するモデルを同時に学習する．生成モデルでは，関係ラベル r をデコーダの各時刻の入力に追加する．

力とする．これらは，任意の三つ組が与えられた時その三つ組が正しい組み合わせか否かを判別するモデルと考えることができる．我々のモデルも上記と同様の定式化を用いて知識補完モデルを定義するが，v_{in} のモデル化が Li ら [5] と異なる．Li らは，フレーズのベクトル表現として，単語ベクトルの平均，LSTM の maxpooling の二種類とシンプルなモデル化を行っている．一方，我々のモデル化では各フレーズのベクトルを LSTM の attention pooling を用いて下記のように定義する．ここで，x^i_j，h^i_j はそれぞれフレーズ t_i の j 番目単語の embedding と LSTM の隠れ層ベクトルを表す．

$$h^i_j = \mathrm{BiLSTM}(x^i_j, h^i_{j-1})(i = 1,2) \tag{2}$$

$$v_i = \sum_{j=1}^{J} \frac{\exp(e_t)}{\sum_{k=1}^{J}\exp(e_k)} h^i_j \tag{3}$$

$$e_k = v^\top \tanh(W h_k) \tag{4}$$

$$v_{12} = \mathrm{Bilinear}(v_1, v_2) \tag{5}$$

$$v_{in} = \mathrm{concat}(v_{12}, v_r) \tag{6}$$

v_{in} は，batch normalization と dropout を行った上で上位の層に受け渡す．各フレーズをベクトル化するための LSTM，単語・関係ラベルの embedding は次に説明するフレーズ知識生成モデルと共有する．

3.2 フレーズ知識生成モデル

次に，フレーズ知識を生成するモデルについて説明する．本研究では，フレーズ知識を生成するためにアテンションベースの Encoder-decoder モデルを用いる．また，エンコーダの LSTM と単語・関係ラベル embedding 部分はフレーズ知識補完モデルと共有する．Encoder-decoder モデルのエンコーダ部分を文やフレーズのベクトル表現として用いる方法については，[3] らが提案しており効果が示されている．本研究でもフレーズのベクトル化部分を Encoder-decoder の情報と共有することでより高品質なフレーズの表現が得られることが期待できる．今回は単語列に加えて，関係ラベル r の情報が存在するため，関係ラベルを考慮した Encoder-decoder モデルを構築する．ここで，入力フレーズの単語列を $X = (x_1, x_2, ..., x_I)$，出力フレーズの単語列を $Y = (y_1, y_2, ..., y_T)$ とすると，Y の出力確率は下記のように定式化できる．

$$p(Y|X,\theta) = \prod_{t=1}^{L} p(y_t|y_{<t}, c_t, r) \tag{7}$$

$$p(y_t|y_{<t}, c_t, r) = g(y_{t-1}, s_t, c_t, r) \tag{8}$$

$$s_t = \mathrm{LSTM}(\mathrm{concat}(v_{y_{t-1}}, v_r), s_{t-1}) \tag{9}$$

ここで，c_t は attention で重みづけられた入力側のコンテキストベクトル，s_t は LSTM の隠れ層を表す．上記に示すように，デコーダの入力として v_r を結合して用いている．このような方法でデコーダ側に追加情報としてラベルを入れる方法については [4] などでも類似の手法が提案されている．パラメータ θ は学習によって求める．

また，今回は Encoder-decoder モデルの学習データとして三つ組データを用いる．三つ組データの場合，どちらのフレーズを入力としても問題ないと考えられるため，入力と出力を入れ替えたデータについても学習を行う．この際，関係ラベルには方向があるため，新たに逆向きのラベル r' を導入する．従って，Encoder-decoder モデルにおいては，関係ラベルの語彙数は元のラベルの語彙数の 2 倍になる．

3.3 学習

3.3.1 損失関数

提案手法では，上記に示した 2 つのモデルの損失関数を同時に考慮しながら学習を行う．具体的には，下記の式を用いて学習を行う．

$$L(\theta) = L_{\mathrm{triple}} + \lambda L_{\mathrm{encdec}} \tag{10}$$

ここで，θ はモデルパラメータであり，L_{triple} はフレーズ知識補完モデルに関する損失関数，L_{encdec} はフレーズ知識生成モデルに関する損失関数を表す．フレーズ知識補完モデルの損失関数 L_{triple} については Li ら [5] の結果から最も精度が良かった binary cross entropy を用いて下記の式で表す．

$$L_{\mathrm{triple}}(\tau, l) \tag{11}$$
$$= -\frac{1}{N}\sum_{n=1}^{N}(l\log\sigma(score(\tau)) + (1-l)\log(1-\sigma(score(\tau))))$$

ここで，τ は三つ組を表す変数，l は正例に対して 1，負例に対して 0 となるバイナリ変数，σ はシグモイド関

数である．上記の定式化は，任意の三つ組 $\tau = \langle t_1, r, t_2 \rangle$ に対して正例のスコアが 1，負例のスコアが 0 に近くなるように学習を行う．

Encoder-decoder の損失関数については，通常の Encoder-decoder モデルと同様に cross entropy 関数を用いて次のように表す．

$$L_{\text{encdec}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{L}\log p(y_t^{(n)}|y_{<t}^{(n)}, c_t^{(n)}, r^{(n)}) \quad (12)$$

ここで，N はデータ数，L は出力側のフレーズ Y の単語数，c_t は入力側のコンテキストベクトル，r は関係ラベルを表す．

3.3.2 負例サンプリング

Binary cross entropy を用いて 2 値分類モデルの学習を行う場合，負例を用意する必要がある．今回扱うフレーズ知識データは，人手アノテーションデータが正例のみであるため，Li ら [5] の研究では負例を自動生成することで学習を行っている．本研究では，Li ら [5] の研究で最も精度が良かったランダムサンプリングを用いて負例の生成を行う．具体的には，それぞれの正例三つ組データ $\tau = \langle t_1, r, t_2 \rangle$ に対して，t_1, t_2, r を 1 つずつランダムに置き換えたデータ $\tau_{neg1} = \langle t_1', r, t_2 \rangle$, $\tau_{neg2} = \langle t_1, r', t_2 \rangle$, $\tau_{neg3} = \langle t_1, r, t_2' \rangle$ を生成する．ランダムにサンプリングされる t_1'，t_2' はそれぞれ学習時のミニバッチ中に出現した候補からサンプリングされ，r' は全ラベル候補の中からサンプリングされる．従って，学習中は，正例 1 つにつき 3 個の負例をサンプリングしながら学習を行う．ただし，負例はフレーズ知識補完モデルのみに用いる．フレーズ知識生成モデルは正しい三つ組から学習を行いたいため，正例の三つ組のみから学習を行う．

3.4 非アノテーションデータの利用

3.4.1 三つ組データの自動抽出

フレーズ知識は，ノードが単語列になっているため十分な学習のためには多量の学習データが必要と考えられる．人手で作成可能なデータ量には限界があるため，人手アノテーションがされていない非アノテーションテキストの情報も効果的にモデルに導入できれば，さらなる精度の向上が期待できる．本研究では，非アノテーションデータの効果を確認するため，非アノテーションテキストとアノテーションデータから学習した学習済みモデルを使って，擬似的な三つ組データを自動生成し効果を確認した．具体的な生成手順を下記に示す．

1. 係り受け解析と接続表現を用いて接続表現を含む文節と係り受け関係にある文節から三つ組候補データを抽出．接続表現は「ので」「ため」など，17 種類の接続表現をあらかじめ定義した．(抽出例：⟨ 免税地域である,ので,品物は格安である ⟩)
2. 抽出した三つ組候補の接続表現を，知識補完モデルで定義した関係ラベル $r \in R$ に置き換えた三つ組を作成する．(例：⟨ 免税地域である,*Antonym*,品物は格安である ⟩)
3. 関係ラベルを置き換えた三つ組に対し，学習済みモデルでスコアリングを行い，閾値以上の候補を擬似データとして抽出する（例：⟨ 免税地域である,*Causes*,品物は格安である ⟩）

本研究では閾値を 0.85 とし，Wikipedia テキストから 312,736 トリプルを抽出した．また，今回は日本語のみ実験を行ったが，英語についても同様にデータを追加することが可能である．

表 1: 実験データの概要

	ConceptNet	Ja
train	100,000	192,714
validation1	1,200	6,889
validation2	1,200	-
test	2,400	6,889
関係ラベル数	34	7
語彙数	21,471	18,119
フレーズ平均単語数	2.02	3.96

3.4.2 自動抽出データを用いた追加学習

自動抽出したデータは，ラベルの空間やデータの性質が人手アノテーションデータとは異なっている．そのため，本研究ではまず非アノテーションデータと人手アノテーションデータを結合したデータを用いて事前学習を行い，その後人手データで再学習（fine tuning）を行った．

4 実験

4.1 実験データ

実験データは，Li ら [5] が公開している ConceptNet (英語) のデータ[1]と，我々が独自にアノテーションした日本語のオープンドメインデータを用いる．表 1 にそれぞれのデータの概要を示す．ConceptNet の方がラベル数が多い．語彙数はいずれも 2 万程度だが，フレーズの平均単語長は日本語データが ConceptNet の倍程度と長くなっている．日本語データに関しては，クラウドソーシングを用いて web 上からクロールした頻出単語に関連する三つ組 $\langle t_1, r, t_2 \rangle$ を作成した．ノイズとなるデータを除去するため，ある作成者が作成した三つ組 $\langle t_1, r, t_2 \rangle$ について，r を隠した状態で別の 3 名のワーカーに適切な r を選択するタスクを行ってもらい，2 人以上が同じラベルを選択したデータのみを使用した．また，test データと validation データに関しては，全員の選択した r が一致したデータからランダムに選択し，それ以外を学習データとした．日本語の test, validation データは，ConceptNet データと同様に正例と負例が 1:1 となるようにデータを作成した．具体的には，まず正例をサンプリングした後，各正例の 3 つ組みの要素 1 つをランダムに選択しテストデータ中の別の要素と置換して作成した．

4.2 評価方法と比較手法

知識補完モデル　ベースラインとして，Li ら [5] の手法 (DNN AVG, DNN LSTM) を用いる．これらは，入力ベクトル v_{in} がそれぞれ単語ベクトルの平均，LSTM の maxpooling をとったものである．ただし，LSTM のモデルでは，t_1 と t_2 を別々にベクトル化して concat した．提案手法に関しては，知識補完モデルを単独で用いた場合（proposed w/o EncDec）と双方を同時に学習した場合（proposed w/ EncDec）の精度評価を行った．評価指標は 2 値判別の正解率を用いた．また，ConceptNet の実験に関しては Li ら [5] と同様に，train データで学習を行い，val1 データでハイパーパラメータの調整，評価を valid2, test データで行った．日本語データも同様に train, validation でパラメータを決定し test で評価をした．

知識生成モデル　ベースラインとして，関係ラベルを用いない Encoder-decoder 単独モデル（EncDec w/o relation single）を用いた．また，関係ラベルを考慮した単独モデル（EncDec w/relation single）と，知識補

[1] http://ttic.uchicago.edu/ kgimpel/commonsense.html

表 2: フレーズ知識補完（2 値分類）の結果

	ConceptNet		Ja
method	valid2	test	test
base (DNN AVG)	0.905	0.923	0.889
base (DNN LSTM)	0.910	0.928	0.877
proposed w/o EncDec	0.911	0.927	0.881
proposed w/ EncDec	0.928	0.938	0.897
proposed w/ EncDec (+data)	0.936	0.948	-
proposed w/ EncDec (+augdata)	-	-	0.897
Li et al [5]	0.913	0.920	-
Li et al (+data)[5]	0.918	0.925	-
human [5]	〜0.950	-	-

完モデルとの同時学習（EncDec w/relation Multi）を比較した．評価は，単語レベルの正解率で評価を行った．

4.3 実験設定

本研究で用いたパラメータについて説明する．LSTM の隠れ層，単語・関係ラベルの embedding は 200 次元，知識補完モデルの中間層の次元を 1000，学習時のバッチサイズは 128，ドロップアウトは 0.2, weight decay は 0.00001 に設定した．また，知識生成モデルのエンコーダには 1 層の双方向 LSTM，デコーダには 1 層の LSTM を用いた．最適化法は SGD を用い，初期学習率は 1.0 に設定し減衰率を 0.5 としてスケジューリングを行った．ただし，非アノテーションデータを用いた学習における fine tuning 時は初期学習率を 0.2 に設定した．loss 関数の λ は 1.0 に固定した．単語，関係ラベルの embedding 初期値は，三つ組学習データと Wikipedia を結合したテキストファイルを基に fastText[1] を用いて事前に計算したベクトルを用いた．

4.4 実験結果

4.4.1 フレーズ知識補完

表 2 に知識補完（2 値分類）の評価結果を示す．下層の行には，Li ら [5] が論文中で報告している中で最も良い精度を示している．ここで+data という行は，学習データを 100k から 300k に増やした場合の評価である．表 2 の結果より，提案手法は既存手法に比べて精度が向上しており，ConceptNet のデータでは従来研究の最高値を超える結果が得られた．特に，データを増やした条件では 2%以上の精度向上が見られ，人間による上限 (〜0.95) にも近づいている．単独モデル (proposed w/o EncDec) と同時学習モデル (proposed w/ EncDec) の比較により，ConceptNet, Japanese データともに，同時学習によって単独モデルよりも良い精度が得られていることがわかる．これは，知識補完問題にとっては生成問題の損失関数が制約として働き，より良いフレーズベクトルが得られたためと考えられる．

4.4.2 フレーズ知識生成

表 3 に，フレーズ知識生成モデルの精度を示す．結果から，ベースラインと関係を考慮した Encoder-decoder モデルで大きな精度差が見られ，関係ラベルを考慮することにより生成の精度が大幅に向上していることがわかる．マルチタスク学習にしたことによる生成モデル側の精度向上はあまり見られないが，教師なしデータを追加することにより生成モデルの精度も向上させることができる．

4.4.3 非アノテーションデータの効果

非アノテーションデータを用いた場合の精度について，表 2, 表 3 の+augdata に示した．非アノテーションデータについては日本語のみで実験を行っている．

表 3: フレーズ知識生成の結果

	ConceptNet		Ja
method	valid2	test	test
EncDec w/o relation (single)	0.535	0.564	0.529
EncDec w/relation (single)	0.598	0.633	0.544
EncDec w/relation (Multi)	0.602	0.633	0.542
EncDec w/relation (Multi + augdata)	-	-	0.553

結果より，補完問題については精度の向上が見られなかったが，生成問題において精度が向上していることがわかる．人手で作成されたアノテーションデータではないノイジーなデータでも生成問題に対しては精度の向上に寄与するということがわかった．補完問題についてもノイジーなデータを加えても悪化はしていないため，擬似的な三つ組データの作り方をさらに工夫することやデータ量を増やすことでさらなる精度の向上も期待できると考える．

5 まとめ

本研究では，任意のフレーズ間の知識を推定する知識補完問題において，フレーズ知識生成モデルを同時に考慮するモデルを提案し，単独モデルで学習するよりも精度が向上することを確認した．ConceptNet の実験では，SOTA を達成した．また，アノテーションされていないテキストも同時に学習に用いることで，さらに精度が向上可能であることを示した．今後は，さらに大規模な教師なしデータの同時活用により，モデルの汎化性能を高める方法を検討する．

参考文献

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on NIPS*, pp. 2787–2795, 2013.

[3] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *Proceedings of the 28th International Conference on NIPS*, pp. 3294–3302. MIT Press, 2015.

[4] Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the ACL*, pp. 994–1003, 2016.

[5] Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the ACL*, pp. 1445–1455, 2016.

[6] Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. World knowledge for reading comprehension: Rare entity prediction with hierarchical lstms using external descriptions. In *Proceedings of the 2017 Conference on EMNLP*, pp. 825–834, September 2017.

[7] Baoxu Shi and Tim Weninger. Open-world knowledge graph completion. *CoRR*, Vol. abs/1711.03438, , 2017.

[8] Qizhe Xie, Xuezhe Ma, Zihang Dai, and Eduard Hovy. An interpretable knowledge transfer model for knowledge base completion. In *Proceedings of the 55th Annual Meeting of the ACL)*, pp. 950–962, July 2017.

[9] Bishan Yang and Tom Mitchell. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the ACL*, pp. 1436–1446, July 2017.