Integrating knowledge graph embeddings to improve mention representation for bridging anaphora resolution

Onkar Pandit, Pascal Denis, Liva Ralaivola

Integrating knowledge graph embeddings to improve mention representation for bridging anaphora resolution

Onkar Pandit[1], **Pascal Denis**[1] **and Liva Ralaivola**[2]

1- MAGNET, Inria Lille - Nord Europe, Villeneuve d'Ascq, France
onkar.pandit@inria.fr, pascal.denis@inria.fr

2- Criteo AI Lab, Paris, France. l.ralaivola@criteo.com

Abstract

Lexical semantics and world knowledge are crucial for interpreting bridging anaphora. Yet, existing computational methods for acquiring and injecting this type of information into bridging resolution systems suffer important limitations. Based on explicit querying of external knowledge bases, earlier approaches are computationally expensive (hence, hardly scalable) and they map the data to be processed into high-dimensional spaces (careful handling of the curse of dimensionality and overfitting has to be in order). In this work, we take a different and principled approach which naturally addresses these issues. Specifically, we convert the external knowledge source (in this case, WordNet) into a graph, and learn embeddings of the graph nodes of low dimension to capture the crucial features of the graph topology and, at the same time, rich semantic information. Once properly identified from the mention text spans, these low dimensional graph node embeddings are combined with distributional text-based embeddings to provide enhanced mention representations. We illustrate the effectiveness of our approach by evaluating it on commonly used datasets, namely ISNotes (Markert et al., 2012) and BASHI (Rösiger, 2018). Our enhanced mention representations yield significant accuracy improvements on both datasets when compared to different standalone text-based mention representations.

1 Introduction

An *anaphor* is an expression whose interpretation depends upon a previous expression in the discourse, an *antecedent*. A *Bridging anaphor* is a special type of anaphor where there is non-identical or associative relation with its antecedent (Clark, 1975), as in the following example:

"*Starbucks* has a new take on the unicorn frappuccino. **One employee** accidentally leaked a picture of the secret new drink."

In this case, the anaphor **One employee** depends on the antecedent *Starbucks* for the complete interpretation and holds non-identical relationship with the antecedent, hence, a bridging anaphor.

We here address the problem of learning from a set of anaphor-antecedent pairs a predictor capable of accurately identify such pairs in unseen texts. More precisely, if bridging resolution comprises two main tasks, *bridging anaphora recognition* and *bridging anaphora resolution*, we solely focus on the task of *bridging anaphora resolution* and assume that bridging anaphor recognition has already been performed.

Semantic information on anaphor-antecedent pairs plays a crucial role in resolving bridging anaphora. Consider again the previous example: if the resolution system has the knowledge that Starbucks is a company and companies have employees, then it is easy to establish the link between them. Standard text-based features either hand-crafted or automatically extracted from word embeddings (Mikolov et al., 2013a, Pennington et al., 2014), are not sufficient for bridging resolution (Hou, 2018b). Earlier systems (Poesio et al., 2004, Lassalle and Denis, 2011) have proposed to extract this information from knowledge bases, the web, or raw text through queries of the form "X *of* Y". The estimated number of occurrences in these sources gives the probability of relations between X and Y. These types of queries were generalized by (Hou et al., 2013) where all queries of the type "X *preposition* Y", i.e. beyond the mere "of" preposition, were considered. However, these approaches extract only shallow features, capturing relations between pair of nodes instead of taking advantage of broader information that is

present in knowledge graphs. Therefore, attempting to extend these strategies to take into account a larger amount of information on mentions may translate into learning problems where the input space is of high dimension, which might be a hurdle when dealing with moderate size datasets – for instance, the datasets that we consider here, i.e ISNotes (Markert et al., 2012) and ARRAU (Uryupina et al., 2019) respectively contain 663 training pairs and 5512 training pairs.

Recently proposed approaches tried to remedy these shortcomings (Hou, 2018b, Hou, 2018a). Hou learned embeddings on the pairs of nouns present in the text which are connected by prepositional or possessive structure (e.g. "X *of* Y"). She creates "pseudo knowledge" by generating these noun-pairs and learn embeddings on these pairs. Her approach is better at capturing fine-grained semantics than *vanilla* word embeddings such as Word2Vec (Mikolov et al., 2013a), Glove (Pennington et al., 2014), etc. however, it still depends on the presence of the required noun-pairs in the corpus. The use of knowledge graphs, either manually or automatically constructed, can alleviate this problem as they contain general semantic and world-knowledge. We empirically demonstrate that embeddings constructed on these graphs indeed provide additional information and complement these text-based embeddings.

In the present work, we propose to use *low-dimensional* graph node embeddings on knowledge graphs to capture semantic information. We use WordNet [1] (Fellbaum, 1998) as a knowledge graph in the experiments, though our approach can be extended to any knowledge graph. We hypothesize that the low-dimensional vectors learned on the nodes of WordNet graph capture lexical semantics such as hypernymy, hyponymy, meronymy, etc. as well as general relatedness between nodes. This way we eliminate the cumbersome task of manually designing features as well as the burden of querying. Moreover, as we shall see, the low dimensionality of the embedding space does not go against its use with small datasets. But obtaining node embeddings for a mention is non-trivial a task, as it requires mapping a potentially ambiguous multi-token expression onto a specific node in the graph (synset in case of WordNet). This entails several key steps, such as: (i) *mention normalization* where the mention is mapped to a standardized form which might be present in the graph, (ii) handling the *absent knowledge* case where the referred entity is unavailable in the knowledge graph and possibly (iii) *sense disambiguation* in the presence of multiple senses for the mention. We propose simple yet effective heuristics to address these issues, as detailed in the coming sections. These knowledge graph embeddings are combined with distributional text-based embeddings to produce improved mention representations.

We address the problem of bridging resolution as a ranking problem, where the trained model assigns a score to anaphor-candidate antecedent pairs, preferring this ranking approach over a classification perspective for it to be less sensitive to class-imbalance, and making it focused on learning relative scores. Specifically, we train a ranking SVM model to predict scores for anaphor-candidate antecedent pairs, an approach that has been successfully applied to the related task of coreference resolution (Rahman and Ng, 2009). We observe that integrating node embeddings with text-based embeddings produces increased accuracy, substantiating the ability of graph node embeddings in capturing the semantic information.

2 Related Work

Bridging anaphora resolution. Earlier approaches (Poesio et al., 1997, Poesio and Vieira, 1998, Poesio et al., 2004, Lassalle and Denis, 2011) put restrictions on the resolution task either by constraining the types of noun-phrases (NP) to be considered as bridging anaphor or by restricting relations between bridging anaphors and antecedent where most of the approaches tackle specific type of anaphor like definite noun-phrases. A pairwise model combining lexical semantic features as well as salience features to perform bridging resolution limited to mereological relations only is studied by (Poesio et al., 2004) on the GNOME corpus. Lexical distance is used as one feature in their approach. WordNet (Fellbaum, 1998) is used to acquire the distance. For a noun head X of an anaphor x and Y of potential antecedent y, the query of the form "X *of* Y" is provided to WordNet. But recall in WordNet is low, so as an alternative, Google API is used to get the distance between anaphor-antecedent. The API yields number of hits, from which lexical distance is calculated. Based on this method (Lassalle and Denis, 2011) developed a system

[1] WordNet is a lexical database and not a knowledge graph in the stricter sense. But, the graph is constructed over it, to be subsequently used as the knowledge graph.

that resolves mereological bridging anaphors in French.

Improving on the previous approach, (Hou et al., 2013) proposed a generic query with all possible prepositions. Their query is formulated as "*X preposition Y*" instead of limiting to *of* preposition. They propose a *global* model as opposed to previous approaches that relied only on *local* features. In this model, they infer links globally instead of choosing from candidate set of the specific anaphor as they argue that the probability of noun phrase (NP) being antecedent increases if it is already antecedent to another anaphor. Their assumption is opposite to the local salience hypothesis of (Sidner, 1979) as the local models indirectly assume that the most salient candidate among the nearest context is the best suitable for antecedent. Rule-based *full bridging resolution* system is proposed in (Hou et al., 2014) where they devised rules for linking anaphors to antecedents. Some of the rules as well as the corresponding features are acquired by querying the knowledge sources, albeit different queries such as a query to get a list of nouns which denote a part of building − *wall, window,* or list of personal relations − *husband, sister*, etc. They also propose a learning-based system by converting the rules into features but observe slight gain.

The work (Hou, 2018b) created word embeddings for bridging (embeddings_PP) by exploring the syntactic structure of noun phrases (NPs) to derive contexts for nouns in the GloVe model. She generalizes previous approaches of querying as her PP context model uses all prepositions for all nouns in big corpora. The deterministic approach proposed in (Hou, 2018a) is the extension to the work done in (Hou, 2018b) which creates new embeddings (embedding_bridging) by combining embeddings_PP and GloVe. Her approach is efficient and solves the scalability and curse of dimensionality issues. But her approach depends on the presence of the NP having a specific syntactic structure so that the algorithm can identify it as "*X preposition Y*". This algorithm misses those anaphor-antecedent pairs which do not possess this structure. The work (Roesiger et al., 2018) uses neural networks trained on the relation classification tasks to get the semantic information between anaphor and antecedent. This information is integrated into the state-of-the-art systems for coreference and bridging resolution. The system fails at capturing broader semantic relations as only six semantic relations are predicted with neural networks, due to this they observe marginal improvement in the bridging resolution.

All the previous works assume that the mentions are detected, i.e., noun phrases are presented and the task is to choose the correct NP as an antecedent. This is discarded in the latest system, BARQA (Hou, 2020). She casts bridging anaphora resolution as a question answering problem where answer produces antecedent for an anaphor. She also pointed out that most of the previous approaches relied only on the features of the antecedent-anaphor ignoring the context around them. However, she ignores any semantic information and relies on BERT (Devlin et al., 2018) architecture to capture both contextual information as well as required common sense knowledge.

Knowledge Graph Embeddings Graph embeddings represent graph (whole or sub-graph) or nodes with the lower dimensional vector. The work (Hamilton et al., 2017) details a generic framework of the commonly used graph embedding algorithms. In recent times, embedding algorithms specifically for knowledge graphs have been proposed − RESCAL (Nickel et al., 2011), DistMult (Yang et al., 2014), ComplEx (Trouillon et al., 2016), HolE (Nickel et al., 2016) learn embeddings for knowledge graph completion, (Bansal et al., 2019) propose A2N neighborhood attention-based technique, (Xu and Li, 2019) embed relations with dihedral groups whereas (Nathani et al., 2019) employ graph attention network to acquire embeddings. In this work, we used WordNet as a knowledge graph so we are interested in the graph node embeddings learned particularly on WordNet (Goikoetxea et al., 2015, Saedi et al., 2018, Kutuzov et al., 2019). Though, (Goikoetxea et al., 2015, Saedi et al., 2018) do not produce embeddings for senses present in WordNet as they encode corresponding words. However, `path2vec` (Kutuzov et al., 2019) produces embeddings for each sense present in WordNet by optimizing graph-based similarity metric. The use of knowledge graph embeddings to infuse common sense knowledge into NLP systems is becoming popular, and our work falls into this category. Language model (Peters et al., 2019), domain-specific natural language inference (NLI) (Sharma et al., 2019), entity disambiguation (Sevgili et al., 2019) have been some of the tasks where graph embeddings have been used. To the best of our knowledge, this is the first work where graph embeddings are used for bridging anaphora resolution.

3 Knowledge-aware mention representation

In this paper, we propose a new, knowledge-aware mention representations for bridging resolution. These representations combine two components: (i) distributional embeddings learned from raw text data, and (ii) graph node embeddings learned from relational data obtained from a knowledge graph. Specifically, the final representation v_m for a mention m is obtained by concatenating the text-based contextual embeddings g_m and the knowledge graph node embeddings h_m: $v_m = [g_m, h_m]$.

For the distributional embeddings g_m, we use off-the-shelf word embeddings such as word2vec (Mikolov et al., 2013a), glove (Pennington et al., 2014), BERT (Devlin et al., 2018), or embeddings_pp (Hou, 2018b). Except for BERT, we average over embeddings of the mention's head word and common nouns appearing in the mention before the head, as mentioned in (Hou, 2018a). With BERT, mention embeddings are obtained by averaging over embeddings of all the words of the mention.

However, obtaining knowledge graph-based embeddings h_m for the mention is a much more challenging task, comprising different steps. Before detailing those steps, we first briefly describe the knowledge graph – WordNet and how we compute node embeddings in the following paragraphs.

Knowledge Graph is a graph with nodes being entities or abstract concepts and edges denoting the relation between them. A node in the knowledge graph can be a real-world entity such as a person, a place, etc. or can be an abstract concept such as a word, a sense, etc. A knowledge graph can be domain-specific (WordNet (Fellbaum, 1998) captures the semantic relation between words and meanings) or open domain (DBpedia (Lehmann et al., 2015) for general-purpose knowledge). The central purpose of knowledge graphs is to store common sense knowledge in a structured format so that machines can easily access it. In this work, we have used WordNet as a knowledge repository but our approach is generic and can be applied with any other knowledge graph.

WordNet (Fellbaum, 1998) primarily consists of *synsets*, i.e., a set of synonyms of words. The *synsets* which refer to the same concept are grouped together giving it a thesaurus-like structure. Each *synset* consists of its definition and small example showing its use in a sentence. The *synsets* are connected with different relations such as synonymy, antonymy, hypernymy, hyponymy, meronymy, etc. In addition to the semantic knowledge, it also includes a bit of common sense knowledge such as real world entities like cities, countries and famous people. However, WordNet stores this knowledge as a database in its basic form, so a graph is constructed based on WordNet for further use. Subsequently, the node embeddings learned on this graph will automatically capture the semantic information associated with the senses.

We briefly discuss different WordNet node embedding algorithms used in our study. We use random walk and neural language model based embeddings (Goikoetxea et al., 2015), matrix factorization based WordNet embeddings (Saedi et al., 2018) and graph-similarity based `path2vec` (Kutuzov et al., 2019) embeddings. The important distinction between these methods is that the first two algorithms (Goikoetxea et al., 2015, Saedi et al., 2018) produce word embeddings and `path2vec` produces embeddings corresponding to each sense present in WordNet. The `path2vec` algorithm naturally encodes WordNet nodes as it actually produces embeddings for senses as opposed to (Goikoetxea et al., 2015, Saedi et al., 2018) algorithms as they conflate all the senses to produce word embeddings instead of generating embeddings for each sense while losing some finer semantic information in the process.

The approach proposed by (Goikoetxea et al., 2015) is based on the well-known neural language model Continuous Bag of Words and Skip-gram (Mikolov et al., 2013b). The main idea is to produce artificial sentences from WordNet and to apply the language models on these sentences to produce word embeddings. For this, they perform random walk starting at any arbitrary vertex in WordNet, then map each WordNet sense to the corresponding word to produce an artificial sentence. Each random walk produces a sentence, repeating this process several times gives a collection of sentences. Finally, this collection of sentences is considered as the corpus for learning word embeddings.

A different approach based on matrix factorization is taken in (Saedi et al., 2018) to produce embeddings. The procedure starts by creating the adjacency matrix M from WordNet graph. The element M_{ij} in the matrix M is set to 1 if there exists any relation between words w_i and w_j. [2] Furthermore,

[2] They also experimented by weighting relations differently (e.g. 1 for hypernymy, hyponymy, antonymy and synonymy, 0.8 for meronymy and holonymy and 0.5 for others) but obtained the best results without weighting.

words which are not connected directly but via other nodes should also have an entry in the matrix, albeit with lower weights than 1. Accordingly, matrix M_G is constructed to get the overall affinity strength between words. In the analytical formulation, M_G can be constructed from the adjacency matrix M as $M_G = (I - \alpha M)^{-1}$ where I is the identity matrix and $0 < \alpha < 1$ decay factor to control the effect of longer paths over shorter ones. Following that, matrix M_G is normalized to reduce the bias towards words which have more number of senses and finally a Principal Component Analysis is applied on it to produce dense word vectors.

The `path2vec` (Kutuzov et al., 2019) learns embeddings based on a pairwise similarity between nodes. The fundamental concept is that pairwise similarity between nodes of the graph should remain the same after their projection in the vector space. The model is flexible enough to consider any user-defined similarity measure while encoding. The objective function is designed to produce such embeddings for nodes which reduce the difference between actual graph-based pairwise similarity and vector similarity. It also preserves the similarity between adjacent nodes. Formally, for the graph $G = (V, E)$ where V, E denote a set of vertices and edges, respectively, the objective is $-$

$$\sum_{(a,b)\in\mathcal{V}} \min_{\mathbf{v}_a,\mathbf{v}_b} \left((\mathbf{v}_a^T\mathbf{v}_b - s(a,b))^2 - \alpha(\mathbf{v}_a^T\mathbf{v}_n + \mathbf{v}_b^T\mathbf{v}_m)\right)$$

where n, m are adjacent nodes of nodes a, b respectively, $s(a, b)$ is the user-defined similarity measure between a, b and $\mathbf{v}_a, \mathbf{v}_b, \mathbf{v}_n, \mathbf{v}_m$ denote the embeddings of a, b, n, m, respectively. To show the ability of their model in adapting to different pairwise similarity measures.

Mention normalization. The first step for being able to align a mention with a particular node in the knowledge base and ultimately its graph embedding, is to convert the mention into a normalized form that can be easily matched. Consider mentions like *the wall*, *one employee*, *beautiful lady* or *the famous scientist Einstein*; none of these can be directly matched to a knowledge graph node (in this case WordNet synset [3]). We propose to normalize them into a single word, respectively to *wall*, *employee*, *lady* and *Einstein*. We design simple rules to normalize mentions. For this, as a first step, we remove articles and commonly used quantifiers like *the, a, an, one, all* etc. from the mention. If we find an entry in the knowledge graph with this modified word then we get the corresponding embedding, otherwise, we go a step further and extract the *head* of the mention and try to obtain embeddings for it. Specifically, we use the parsed tree of the mention and Collins' head finder algorithm (Collins, 2003) to get the head.

Absence of Knowledge. Even after mention normalization, it might still be possible that a mention cannot be aligned with a node in the knowledge graph, simply because some entities are not present therein. This leads to the unavailability of the corresponding node embeddings. We use zero vector of the same dimensions to resolve these cases where node embeddings are absent.

Sense disambiguation The knowledge graph may contain multiple concepts or senses for a given entity. This is the case in all the knowledge graphs. The reason is that the same word has many senses or refer to different real world entities. For example, the word *bank* can refer to *a financial institution* or *the land alongside the river*, the entity *Michael Jordan* can refer to *the scientist* or *the basketball player*. Due to this ambiguity, there are multiple node embeddings for the same mention as they capture entirely different concepts [4]. However, recognizing the correct sense is crucial to get accurate embedding. We explore two simple heuristics to tackle the issue of multiple senses of an entity $-$ 1. *Lesk* (Lesk, 1986) algorithm to get the correct sense of the mention depending on the context, as in the discourse, the meaning of the mention depends on the context in which it is uttered. 2. Unweighted average over embeddings of all the senses of the mention.

4 Ranking Model

Let $\mathcal{D}$ be the given document containing $\mathcal{M} = \{m_1, m_2, \cdots, m_{n_m}\}$, n_m number of mentions. Let $\mathcal{A} = \{a_1, a_2, \cdots, a_{n_a}\}$ denote the set of all anaphors and $\mathcal{A} \subset \mathcal{M}$. Let a be any anaphor in the set $\mathcal{A}$

[3] In case of WordNet embeddings from (Goikoetxea et al., 2015, Saedi et al., 2018), normalized mention is mapped to words.

[4] This difficulty does not arise in the cases where embeddings are learned for words instead of senses (Goikoetxea et al., 2015, Saedi et al., 2018). But, the problem is prevalent for node embeddings learned for actual nodes of the graph.

and j be its position in the set $\mathcal{M}$, then E_a be the set of candidate antecedents for a which is defined as $E_a = \{m_i : m_i \in \mathcal{M}, i < j\}$. Let T_a and F_a be the set of true antecedents and false candidate antecedents of a such that $T_a \cup F_a = E_a, T_a \cap F_a = \emptyset$. Let each anaphor a is represented with the feature vector v_a and candidate antecedent e represented with v_e where $e \in E_a$. Then the goal is to predict score $s(v_a, v_e)$ between anaphor a and candidate antecedent e. The score denotes the possibility of anaphor a having bridging relation with the candidate antecedent e, so a higher score denotes a higher chance of e being true antecedent.

The model is trained to reduce the ranking loss calculated based on the scores obtained between anaphor-candidate antecedents. The ranking strategy is fairly obvious – for an anaphor a high scoring candidate antecedent from E_a is ranked higher than the low scoring one. Let this prediction ranking strategy be r' and true ranking is given by r^*. For a candidate antecedent, if predicted rank is not the same as true rank then it is called discordant candidate, otherwise concordant. The difference between true and predicted ranking strategy can be measured with Kendall's rank correlation coefficient – τ. Formally, concordant C, discordant D candidates and τ are calculated as –

$$C = \sum_{(t,f)\in(T_a \times F_a)} \mathbb{I}_{s(v_a,v_t)>s(v_a,v_f)} \ , \quad D = |T_a \times F_a| - C \quad and \quad \tau(r^*, r') = \frac{C-D}{C+D}$$

where $\mathbb{I}$ is an indicator function which takes value 1 if $s(v_a, v_t) > s(v_a, v_f)$ else 0 and $|\cdot|$ denotes cardinality of the set. The empirical ranking loss (Joachims, 2002) captures the number of wrongly predicted ranks which is given as –

$$\mathcal{L} = \frac{1}{n_a} \sum_{i=1}^{n_a} -\tau(r_i^*, r_i')$$

Inference We consider all the anaphors in the test document separately. For each anaphor, we consider all previously occurring mentions as candidate antecedents [5] and find out the compatibility score for each anaphor-candidate antecedent pair with the above ranking model. We apply best first strategy to choose the most appropriate antecedent from the list of candidate antecedents. In this strategy, the highest scoring pair is selected as anaphor-antecedent pair. Formally, let a be any anaphor and E_a denote a set of candidate antecedents for a. Let $s(a, e)$ be the score between a and e where $e \in E_a$. Let $\hat{e}_a$ be the predicted antecedent of a which is given by - $\hat{e}_a = argmax_{e \in E_a}\ s(a, e)$

5 Experimental Setup

Data We used ISNotes (Markert et al., 2012) and BASHI (Rösiger, 2018) datasets for experiments. ISNotes and BASHI consist of 50 different OntoNotes documents, containing 663 and 459 anaphors, respectively. BASHI dataset annotates *comparative* anaphors as bridging anaphors which are 115 in numbers, remaining are *referential* anaphors. Following the setup from (Hou, 2020), we only consider 344 referential bridging anaphors in this work as well from the BASHI dataset. In the experiments, we implemented nested cross-validation to select the best hyperparameter combination. The setup is – first we make 10 sets of train and test documents containing 45 and 5 documents respectively with 10-fold division. Then at each fold, 45 training documents are further divided into 5 sets of 36-9 actual training and development documents. Each hyperparameter combination is trained on these 5-sets and evaluated. The highest averaged accuracy over the 5-sets of development documents gives the best hyperparameter combination. Once the best hyperparameter setting is obtained the SVM model is re-trained over 45 documents (36+9). For each fold number of accurately linked anaphors is calculated. The accurately predicted number of anaphors over each fold is added to get the total number of accurately linked anaphors from the complete dataset. Thus, the system is evaluated by the accuracy of predicted pairs (Hou, 2020).

For the training data, we have positive samples where we know true anaphor-antecedent pairs but no negative samples. We generate these pairs by considering all the noun phrases (NPs) which occur

[5] In ISNotes dataset 71% of anaphors have antecedent either in the previous two sentences or the first sentence of the document. So, mentions only from the previous two sentences and the first sentence are considered as candidate antecedents. We apply the same strategy for BASHI dataset as well.

Data		Our Experiments						SOTA	
		WV	GV	BE	EP	BEP	–	SYS	ACC
ISNotes	–	25.94	27.60	32.87	31.08	37.10	-	PMIII	36.35
	+ PL	26.40	28.61	34.39	31.81	43.87*	20.06	MMII	41.32
	+ PA	24.74	30.92	33.18	33.24	39.82*	19.53	EB	39.52
	+ RW	27.75	27.6	34.12	33.24	**46.30***	22.06	MMEB	46.46
	+ WNV	21.71	25.13	31.69	26.80	33.28	17.64	BARQA	50.08
BASHI	–	22.92	17.48	31.23	28.51	33.52	-	PMIII	-
	+ PL	30.95	21.49	35.53	29.26	36.68*	16.44	MMII	-
	+ PA	24.07	19.2	35.24	29.48	**38.94***	17.62	EB	29.94
	+ RW	26.64	18.91	34.38	28.91	38.83*	15.75	MMEB	-
	+ WNV	20.92	18.05	26.36	21.20	27.80	12.97	BARQA	38.66

Table 1: Results of our experiments and state-of-the-art models over two datasets – ISNotes and BASHI. In our experiments section, we present results for different text-based embeddings – word2vec (WV), glove (GV), BERT (BE), embeddings_pp (EP), BERT + embeddings_pp (BEP) and the last column – shows the absence of text-based embeddings. Also, in each row, WordNet node embeddings based on different algorithms, except the first row, are added – path2vec with Lesk (PL), path2vec with averaged senses (PA), random walk based (RW) and WordNet embeddings (WNV). The other section of the table – SOTA, shows results with previously proposed systems – Pairwise Model III (PMIII), MLN model II (MMII) (Hou et al., 2013), embeddings_bridging (EB) (Hou, 2018a), the combination of embeddings_bridging and MLN model (MMEB) and the latest system, BARQA (Hou, 2020). The results with $*$ are statistically signficant in comparison to the results based only on text embeddings with p-value $< 10^{-4}$ with McNemar's test and Wilcoxon signed-rank test.

before the anaphor in the window of some fixed number of sentences. All the mention pairs which do not hold bridging relations are considered as negative samples for training. Similarly at the test time, for an anaphor, all the previous mentions in the fixed window size are considered as candidate antecedents.

Implementation We obtained pre-trained word2vec (Mikolov et al., 2013a), Glove (Pennington et al., 2014), BERT (Devlin et al., 2018) and embeddings_pp (Hou, 2018b) embeddings. We used spanBERT (Joshi et al., 2020) embeddings in our experiments as it gave better results in (Hou, 2020). Also, we used pre-trained WordNet embeddings provided by respective authors of (Goikoetxea et al., 2015, Saedi et al., 2018, Kutuzov et al., 2019). In the case of `path2vec` (Kutuzov et al., 2019), embeddings learned with different similarity measures such as – Leacock-Chodorow similarities (Leacock and Chodorow, 1998); Jiang-Conrath similarities (Jiang and Conrath, 1997); Wu-Palmer similarities (Wu and Palmer, 1994); and Shortest path similarities (Lebichot et al., 2018), are provided. We experimented with all the four similarity measures and found out that the shortest path based similarity measure produced better results most of the time, so we have used those embeddings in our experiments. We used python implementation of *Lesk* algorithm from *nltk*[6] library to select the best sense from multiple senses of the mention. Two sentences previous to mention and two sentences after the mention, including the sentence in which the mention occurs, are given to this algorithm as a context for a mention.

Both anaphor and candidate antecedent's embeddings are obtained as mentioned above, afterwards, element-wise product of these vectors is provided to the ranking SVM. We also did preliminary experiments with the concatenation of the vectors but element-wise product gave better results. We used SVM^{rank} (Joachims, 2006) implementation for our experiments. In the experiments with SVM, we did grid search over C = 0.001,0.01,0.1,1, 10,100 with the use of *linear* kernel. We also use *random fourier features (rff)* trick proposed by (Rahimi and Recht, 2008) to approximate non-linear kernels. We found, use of non-linear kernels slightly improved results in comparison to linear kernels so reported only those

[6] `https://www.nltk.org/_modules/nltk/wsd.html`

results. We also varied different widow sizes of sentences − 2,3,4 and all previous sentences, in addition to NPs from the first sentence (salience), to get candidate antecedents for an anaphor. Out of these settings, the window size of 2 and salience have yielded the best results which are reported here.

6 Results

Comparison between distributional and graph embeddings is shown in Table 1 in our experiments section. The first row corresponding to ISNotes and BASHI dataset shows results with only text-based embeddings. We observe that on both the datasets the best performance is obtained with the use of BERT embeddings showing the efficacy of these embeddings when only one type of text-based embeddings is used. It shows that the context of the mention plays important role in resolving bridging anaphora. The second best scores are obtained with embeddings_pp which are specially designed embeddings for the task. We also observe further improvement in the results when two best performing text-based embeddings − BERT and embeddings_pp are combined (noted as BEP in the Table)[7].

The following rows (2-4) of Table 1 show the results obtained with the addition of WordNet information with different embeddings algorithms − `path2vec` (Kutuzov et al., 2019)(PL and PA), random walk based embeddings (Goikoetxea et al., 2015) (RW) and WordNet embeddings (Saedi et al., 2018)(WNV). The results from these rows in comparison with the result from the first row prove the effectiveness of the external information and substantiates our claims [8]. Interestingly, it also shows that BERT though trained on a huge unlabelled corpus is not inherently efficient at capturing common sense knowledge required for bridging anaphora resolution. Though, it has been competitive at capturing relational knowledge required for other nlp tasks like question answering (Petroni et al., 2019). Moreover, external information seems to be complementing embeddings_pp embeddings which are custom tailored for bridging tasks, further consolidating our claims. We compare results from `path2vec` Lesk (PL) with `path2vec` average (PA) to see which strategy of disambiguation is effective. But the observations are not conclusive, as in some cases performance with the use of averaging strategy is better than choosing the best sense with Lesk. The reason is that Lesk is a naive algorithm which considers overlapping words in the context to get the best sense. Further, in each row of the second last column of the table, results obtained by combining external information with BERT embeddings and embeddings_pp show that even the best performing text-based embeddings can still benefit from the external information.

Comparison between different WordNet embeddings We first examine the effectiveness of external knowledge without any text-based embeddings. These scores are noted in the last column of our experiments section against each WordNet graph node embeddings. The lower scores in this column in comparison with text-based embeddings reveal that the features learned with WordNet embeddings are not sufficient and should be complemented with the contextual features. This observation further substantiates our observation of higher scores with BERT embeddings showing the importance of context (Table 1, the first row). Further, we consider results from averaged embedding over senses (PA) for comparing `path2vec` with the other two embeddings as it is the closest analogous setting to correlate. This comparison shows, there is no best algorithm amongst these WordNet embeddings as sometimes we get better results with `path2vec` and sometimes with random walk based embeddings. This result is surprising as even after losing some semantic information, RW produces competent results compared to `path2vec`. This might be happening because of errors in sense disambiguation with `path2vec`.

Comparison with previous studies The results of different state-of-the-art systems on both the datasets are presented in SOTA section of Table 1. These results are obtained from Hou's latest work (Hou, 2020). In BARQA (Hou, 2020), mentions are also detected in her model, so we considered results where gold mentions are considered for the equal comparison. We observe that, on ISNotes dataset, our model's performance is better than rule-based approaches from Pairwise Model III and MLN model II (Hou et al., 2013), embeddings_bridging based deterministic approach from (Hou, 2018a) and competitive

[7] We combine BERT and embeddings_pp embeddings by concatenating both the vectors

[8] Except with the addition of WordNet embeddings (WNV) as results with WNV are mostly inferior in comparison with only text-based embeddings. Lower coverage for WNV, around 65% as opposed to 90% for the other two embeddings as only 60,000 words were present in pre-trained WNV embeddings, might be the possible reason. Also, the vector dimension is significantly higher − 850 in comparison to 300 for the other two.

Mention Mapping Error		Mention Sense Selection	
Mention	Normalized Mention	Mention	Selected Sense
Los Angeles, Cali.	Angeles	[...] future generations of memory **chips**	electronic equipment
Hong Kong	Kong	The move by the coalition of political **parties** [...]	organization
U.S.S.R	U.S.S.R	[...] when the rising Orange River threatened to swamp the **course** [...]	route
IBM	IBM	[...] U.S. industry to head off **the Japanese**, who now dominate [...]	language
politburo member Joachim Herrman	Herrman	[...] potential investors at race **tracks** [...]	magnetic paths
U.S. district judge Jack B. Weinstein	Weinstein	The Thoroughbred Owners and Breeders **Association** [...]	a group of organisms

Table 2: **Mention Mapping Error** lists examples of mentions for which no entry is found in WordNet after normalization. The first three mentions are not found because of normalization error but the next three entities are not present in WordNet. **Mention Sense Selection** notes a few mentions and their senses selected by Lesk algorithm. For the first three mentions, Lesk disambiguates correctly but fails in the next three. The correct senses of the last three are *Japanese people*, *racecourse* and *organization*, respectively.

in comparison with the combination of MLN model and embeddings_bridging but lags to BARQA model. The reason might be that MLN model combines hand-crafted rules in addition to carefully crafted embeddings. On the other hand, BARQA system is trained on additional data obtained by forming quasi-bridging pairs. However, with BASHI dataset we observe best results, as the model achieves significant gains in comparison with embeddings_bridging and moderate gains against BARQA.

7 Error Analysis

7.1 Mention normalization and sense disambiguation

ISNotes dataset contains 663 anaphors and combining those with candidate antecedents of each anaphor we get more than 9500 mentions out of which 10% of mentions can not be mapped to WordNet entries. The situation is similar in the case of BASHI dataset as around 8% of the 5933 mentions can not be mapped to WordNet entries.

We analyze cases where normalized mention is failed to map to any sense in WordNet. There are broadly two reasons for not getting WordNet entry for the mention − 1. Normalization error 2. Inherent limitations of WordNet. We note down some of the examples from each category in the Table 2. The first three mentions are wrongly normalized as Los Angeles to Angeles and Hong Kong to Kong, otherwise, both the cities are present in WordNet. The cases like U.S.S.R shows limitations of our simple normalization approach, the normalization should map U.S.S.R to Soviet Russia which is present in WordNet. The other three examples show the inherent limitations of WordNet as those entities are absent from WordNet.

WordNet contains multiple senses for a given word because of which we get on an average 7 senses for the given mention. We used a simple *Lesk* algorithm for disambiguation which takes into account the context of the normalized mention to determine the correct sense. We present some examples of disambiguation with Lesk in Table 2. It correctly disambiguates in the first three examples but fails for the following three. This is because of the count of overlapping words between sense's context and definition in WordNet. For example, the last example contains words like blood, breeder in the context because of

which it selects sense as *a group of organisms* and not an *organization*.

7.2 Anaphor-antecedent predictions

We analyze a few anaphor-antecedent pairs which were identified incorrectly with BERT-based mention representation but with the addition of WordNet information, we were able to correct it. The underlined and bold lettered phrases denote antecedent and anaphor, respectively.

(1). Staar Surgical Co.'s board said that it has removed Thomas R. Waggoner [...]. [..] that John R. Ford resigned as **a director**, and that Mr. Wolf was named a member of the board.

(2). So far this year, rising demand for OPEC oil and production restraint by some members have kept **prices** firm despite rampant cheating by others.

(3). One building was upgraded to red status while people were taking things out, and a resident who was not allowed to go back inside called up **the stairs** to his girlfriend, telling her to keep [...].

WordNet contains relations where *company* and *director* are related where director works at company. The *OPEC oil* is stored as a corporation which in turn is related to *prices* and *stairs* are part of *building*. This information is present in WordNet which has been used for resolving these pairs as opposed to relying only on the textual information in case of mention representation only with BERT.

Conversely, we also observed a few pairs where the addition of extra information has been detrimental. The italic faced phrase is the selected antecedent with WordNet based system but without WordNet correct antecedent (shown with underline) was selected for boldfaced anaphor.

(4).Within the same nine months, *News Corp.* [...]. Meanwhile, American Health Partners, publisher of American Health magazine, is deep in debt, and Owen Lipstein, **founder**[...].

(5)[...] *the magnificent dunes where the Namib Desert meets the Atlantic Ocean* [...]Since this treasure chest [...] up a diamond from **the sand**.

(6).The space shuttle Atlantis landed [...] that dispatched *the Jupiter - bound Galileo space probe*. **The five astronauts** returned [...].

Example 4, *News Corporation* is closer to **founder** than Partners as head word is Partners for the long phrase. Thus, the system assigns higher scores to wrong candidate antecedent. Similarly, in example 5, the *dunes* are closer to **sand** than treasure chest. In the example 6, WordNet contains Atalantis as legendary island and not as a space shuttle thus **astronauts** is closer to space probe than island, thus receiving a higher score than the correct antecedent. These mistakes can be attributed to the process of normalizing mentions as well as limitations of WordNet. Interestingly, these examples show the inadequacy of BERT in capturing the *partOf* relation but efficacy of capturing some form of relatedness of the terms.

8 Conclusion

We presented a simple approach of incorporating external semantic knowledge for bridging anaphora resolution. We combined contextual embeddings learned only on the text with the knowledge graph node embeddings. We establish the potency of knowledge graph embeddings with the experiments with the use of different WordNet graph embeddings on the ISNotes and BASHI datasets. Though we apply a simplistic approach to solve mention normalization, absent knowledge resolution and sense disambiguation to obtain node embeddings, we achieve competitive results on both the datasets. Moreover, this study opens up further investigation into the design of sophisticated methods to incorporate knowledge graph embeddings for bridging anaphora resolution such as improved mention normalization and sense disambiguation, incorporating knowledge from multiple knowledge sources.

Acknowledgements

We thank the three anonymous reviewers for their comments and feedback. This work was supported by the French National Research Agency via grant no ANR-16-CE33-0011-01 as well as by CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020.

References

Trapit Bansal, Da-Cheng Juan, Sujith Ravi, and Andrew McCallum. 2019. A2N: Attending to neighbors for knowledge graph inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4387–4392, Florence, Italy, July. Association for Computational Linguistics.

Herbert H. Clark. 1975. Bridging. In *Theoretical Issues in Natural Language Processing*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1434–1439, Denver, Colorado, May–June. Association for Computational Linguistics.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. cite arxiv:1709.05584Comment: Published in the IEEE Data Engineering Bulletin, September 2017; version with minor corrections.

Yufang Hou, Katja Markert, and Michael Strube. 2013. Global inference for bridging anaphora resolution. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 907–917, Atlanta, Georgia, June. Association for Computational Linguistics.

Yufang Hou, Katja Markert, and Michael Strube. 2014. A rule-based system for unrestricted bridging resolution: Recognizing bridging anaphora and finding links to antecedents. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2082–2093, Doha, Qatar, October. Association for Computational Linguistics.

Yufang Hou. 2018a. A deterministic algorithm for bridging anaphora resolution. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1938–1948, Brussels, Belgium, October-November. Association for Computational Linguistics.

Yufang Hou. 2018b. Enhanced word representations for bridging anaphora resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 1–7, New Orleans, Louisiana, June. Association for Computational Linguistics.

Yufang Hou. 2020. Bridging anaphora resolution as question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1428–1438, Online, July. Association for Computational Linguistics.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference*, pages 19–33, Taipei, Taiwan, August. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 02, page 133142, New York, NY, USA. Association for Computing Machinery.

Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 06, page 217226, New York, NY, USA. Association for Computing Machinery.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Andrey Kutuzov, Mohammad Dorgham, Oleksiy Oliynyk, Chris Biemann, and Alexander Panchenko. 2019. Learning graph embeddings from WordNet-based similarity measures. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 125–135, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Emmanuel Lassalle and Pascal Denis. 2011. Leveraging different meronym discovery methods for bridging resolution in french. In Iris Hendrickx, Sobha Lalitha Devi, António Horta Branco, and Ruslan Mitkov, editors, *Anaphora Processing and Applications - 8th Discourse Anaphora and Anaphor Resolution Colloquium, DAARC 2011, Faro, Portugal, October 6-7, 2011. Revised Selected Papers*, volume 7099 of *Lecture Notes in Computer Science*, pages 35–46. Springer.

Claudia Leacock and Martin Chodorow, 1998. *Combining Local Context and WordNet Similarity for Word Sense Identification*, volume 49, pages 265–. 01.

Bertrand Lebichot, Guillaume Guex, Ilkka Kivimäki, and Marco Saerens. 2018. A constrained randomized shortest-paths framework for optimal exploration. *CoRR*, abs/1807.04551.

Jens Lehmann, Robert Isele, Max Jakob, A. Jentzsch, D. Kontokostas, Pablo N. Mendes, S. Hellmann, M. Morsey, Patrick van Kleef, S. Auer, and C. Bizer. 2015. Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6:167–195.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC 86, page 2426, New York, NY, USA. Association for Computing Machinery.

Katja Markert, Yufang Hou, and Michael Strube. 2012. Collective classification for fine-grained information status. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 795–804, Jeju Island, Korea, July. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June. Association for Computational Linguistics.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. 2019. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, Florence, Italy, July. Association for Computational Linguistics.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML11, page 809816, Madison, WI, USA. Omnipress.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI16, page 19551961. AAAI Press.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, November. Association for Computational Linguistics.

Massimo Poesio and Renata Vieira. 1998. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216.

Massimo Poesio, Renata Vieira, and Simone Teufel. 1997. Resolving bridging references in unrestricted text. In *Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*.

Massimo Poesio, Rahul Mehta, Axel Maroudas, and Janet Hitzeman. 2004. Learning to resolve bridging references. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 143–150, Barcelona, Spain, July.

Ali Rahimi and Benjamin Recht. 2008. Random features for large-scale kernel machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc.

Altaf Rahman and Vincent Ng. 2009. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore, August. Association for Computational Linguistics.

Ina Roesiger, Maximilian Köper, Kim Anh Nguyen, and Sabine Schulte im Walde. 2018. Integrating predictions from neural-network relation classifiers into coreference and bridging resolution. In *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 44–49, New Orleans, Louisiana, June. Association for Computational Linguistics.

Ina Rösiger. 2018. BASHI: A corpus of wall street journal articles annotated with bridging links. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Chakaveh Saedi, António Branco, João António Rodrigues, and João Silva. 2018. WordNet embeddings. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 122–131, Melbourne, Australia, July. Association for Computational Linguistics.

Özge Sevgili, Alexander Panchenko, and Chris Biemann. 2019. Improving neural entity disambiguation with graph embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 315–322, Florence, Italy, July. Association for Computational Linguistics.

Soumya Sharma, Bishal Santra, Abhik Jana, Santosh Tokala, Niloy Ganguly, and Pawan Goyal. 2019. Incorporating domain knowledge into medical NLI using knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6092–6097, Hong Kong, China, November. Association for Computational Linguistics.

Candace L. Sidner. 1979. *Towards a computational theory of definite anaphora comprehension in English discourse*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Tho Trouillon, Johannes Welbl, Sebastian Riedel, ric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction.

Olga Uryupina, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J. Rodriguez, and Massimo Poesio. 2019. Annotating a broad range of anaphoric phenomena, in a variety of genres: The arrau corpus. *Natural Language Engineering*, pages 1–34, 05.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ACL 94, page 133138, USA. Association for Computational Linguistics.

Canran Xu and Ruijiang Li. 2019. Relation embedding with dihedral group in knowledge graph. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 263–272, Florence, Italy, July. Association for Computational Linguistics.

Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases.