Canonicalizing Knowledge Bases for Recruitment Domain

Nausheen Fatma^{1(✉)}, Vijay Choudhary¹, Niharika Sachdeva¹,
and Nitendra Rajput²

¹ Naukri.com (Info Edge), Noida, India

{nausheen.fatma, vijay.choudhary, niharika.sachdeva}@naukri.com

² Mastercard, New Delhi, India

nitendra@acm.org

Abstract. Online recruitment industry holds large amount of user-generated content in the form of job postings, resumes etc. This content finds its way in the knowledge bases (KB) causing duplicate and non-standard representations of entities (like company names, institute names, designations, skills etc.) These non-standard entity representations impact various applications such as search, recommendations and information retrieval. Therefore, KB canonicalization i.e., mapping multiple references of same entities into unique clusters is imperative for online recruitment platforms. Research suggests various approaches that use enriched semantic context or external context (from sources like Freebase) to perform KB Canonicalization. In fields where such external sources of context do not exist the problem remains challenging. To address these challenges, we propose a novel deep Siamese architecture with character-based attention and word embeddings that (a) estimates pairwise similarity between all entity mentions, and (b) then uses these similarity (scores) to create canonical clusters representing unique entity in the KB. Our experiments on recruitment domain dataset comprising of 62,288 unique entities of various types such as companies, institutes, skills, and designations demonstrate the effectiveness of our approach. We also provide insights on different network architectures, each of which encapsulate a different set of variation while performing canonicalization.

1 Introduction

User-generated content from candidate CVs, recruiter job posts, and company descriptions is one of the largest source of data on online recruitment platforms (such as LinkedIn, Indeed, Glassdoor, etc.). This content includes diverse entities from recruitment domain such as company names, institute names, skills, and designations, which are extracted and fed to Knowledge Bases (KBs) maintained by these companies. As this content is user generated, multiple variations of these entities find their way into the KBs. Often a company name or an

This work was done when Nitendra was employed at Info Edge.

© Springer Nature Switzerland AG 2020

H. W. Lauw et al. (Eds.): PAKDD 2020, LNAI 12085, pp. 500–513, 2020.

https://doi.org/10.1007/978-3-030-47436-2_38

institute name could have hundreds of variations in the KBs. These name variations lack standard nomenclature or representation, thereby resulting in identical and noisy records for the same entity. For example, KB may have multiple representations named *warner bros.*, *warner brothers pictures* and *warner bros. entertainment inc.* for the same real world entity *Warner Brothers Studios*. Such non-standard representations of named entities when used in applications such as search and recommendations result in sub-optimal information retrieval. Therefore, KB canonicalization i.e, mapping multiple references of same entities into unique clusters is imperative for many applications in the recruitment domain. In this work, we focus on recognizing varied representations of an entity in the KB and group them to form a cluster referring to a unique entity [8, 16, 17]. To canonicalize these varied representations, we address various challenges resulting from previously unseen or out-of-knowledge-base (OOKB) entities, location variations (e.g., *Google Singapore* vs. *Google India* or *Cal State University Los Angeles* vs. *Calif. State Univ LA*), semantic variations (*steam boiler operator* vs. *steam turbine operator*), unconventional abbreviation (*Integrated Electronics* vs. *Intel*), misspellings, unrelated or irrelevant entries (*Not-applicable* or *freelancer*).

Various research works have focused on entity canonicalization [8, 30]. However, these approaches cannot be directly applied to domain-specific KBs due to: (a) low coverage of domain-specific entities in open KB. For example, only 10% of entities in our data were present in Freebase. So research that relies on using Open KBs to help in disambiguation [7, 13] cannot be used for the recruitment domain; (b) Handcrafted features (e.g. lexical, linguistic, and semantic) are limiting as they do not scale to capture the volume of variations. Such approaches have been known to be cost in-efficient and sub-optimal [10, 27]. These limitations warrant for new solutions to enable KB canonicalization in the recruitment domain. We focus on canonicalization with the constraint that there is no external source of information or handcrafted features from open KBs available to us. We build upon [6] where we demonstrated the effectiveness of a deep Siamese BiLSTM network towards learning useful representations from character sequences. In this paper, we use attention mechanism and word embeddings to define a similarity function, thus, capturing similarity both at syntactic and semantic levels in a more focused manner. While in [6], we had used a proprietary KB consisting of 50 million CVs and canonicalized the organization names only, in this paper we also validate the efficacy of our approach for additional entity types as well as open source datasets including ESCO¹ and DBpedia [1], thus demonstrating the generalizability of this solution.

Our approach for finding pairwise similarity achieves state-of-the art performance on all entity types surpassing the baselines widely relied upon in the literature. These results demonstrate the effectiveness and robustness of our approach on real-world data. The main contribution of our work are:

- We suggest a deep learning (BiLSTM with attention mechanism) module for pair-wise similarity estimation followed by Hierarchical Agglomerative Clustering (HAC). Our proposed architecture scales well for well-known as well

¹ <https://ec.europa.eu/esco/portal>.

as unseen and rare entities that have limited or no deducting information available for identifying canonical forms in the open KBs.

- We demonstrate the usefulness of capturing sequence patterns at character and word level for calculating pairwise similarity score and compare it to state-of-the-art similarity measures for KB canonicalization. We provide insights on various network architecture choices for finding pairwise similarity.
- We release our created entity dataset for evaluating the pairwise similarity on both the public datasets (ESCO and DBpedia).

2 Related Work

Our work is related to research on KB Canonicalization, entity linking and clustering. In this section, we discuss the relevance these research works to our paper.

Various research works focus on **canonicalizing entities in the open KBs** including NELL, TextRunner, and ReVerb45K [8, 27]. These works demonstrate the efficacy of employing string similarity and clustering approaches to identify canonical forms of entities or relations in the Open KB [6, 8, 30]. Such approaches, however, recommend handcrafted features which result in sub-optimal solution as they capture limited similarity (semantic or syntactic) [10, 27]. Research on KB generation from multiple sources such as Freebase and Wikipedia also discusses entity canonicalization solutions [27]. These works utilize side information or external hierarchies for canonicalization. Such information is often missing for entities in a domain-specific setting like ours causing limited scalability for emerging or new OOKB entities [12, 21].

Entity Linking research maps entity variations in Open or proprietary KBs to an existing inventory of unique entities [16, 17, 26]. These research works explore candidate entity set generation followed by ranking entities in these candidate set and using this knowledge to predict relatable entities [23]. In [30], authors discuss the creation of mappings from surface forms to possible entities for listing candidate entities. Some systems explore generating a candidate set using string matching/similarity or identifying associated pages through search APIs [5, 8, 30]. Research demonstrates the efficacy of deep neural networks to enhance the sparse features used in Entity Linking. The enhancements improve the generalizability of these features for canonicalization tasks. For instance, feedforward networks have been used on bag-of-words to enhance the entity context or entity class information [11, 24]. Few others propose CNN architecture to capture information about entities at various granularities [7, 13]. However, these approaches do not scale for new or emerging entities and capture limited semantic similarities.

Clustering Approaches: After evaluating the similarity between entity pairs; these entities need to be clustered into high-quality coherent groups representing unique entities. Researchers have widely explored various methodologies to cluster equivalent entities for canonicalization [2, 8]. These clustering approaches are based on partitions, hierarchy, or graphical structures of the underlying text [15, 20, 29]. These methods perform clustering on the hand-crafted similarity scores generated between different pairs of named-entities. Among these

Fig. 1. Entity canonicalization solution (ECS) architecture

methods, Hierarchical Agglomerative Clustering (HAC) is the most extensively used in the literature [8, 27].

We investigate mechanisms for a) similarity using deep Siamese BiLSTM networks with b) clustering on similarity graph for generating high-quality groups of canonicalized entities. We believe, our work will help develop real-time system robust to existing and new or emerging entities with missing information or incomplete entity types in the KB. We use character level embeddings to handle various challenges such as spelling errors and out-of-knowledge-base (OOKB)/emerging entities in the user-generated data.

3 Entity Canonicalization Solution (ECS)

In this section, we describe terminology and our proposed solution ECS.

3.1 Terminology

We define E as the space consisting of all entity variations in our knowledge base. In this space, each entity cluster E_x has n variations represented as $\{e_x^1, e_x^2, \dots, e_x^n\}$. First, to calculate the similarity score, we build a pair-wise similarity function f for every entity pair (e_x^i, e_y^j) in the space E . An entity pair may or may not belong to the same cluster. Hence, we model the function f , such that, $f(e_x^i, e_y^j) = 1 \forall (x, y, i, j)$ where $(x=y)$ and 0 otherwise. Next, we perform clustering using the pair-wise scores estimated by f into groups representing unique entities. Similar nomenclature was used in our previous solution [6].

3.2 Pairwise Similarity Model

We propose a deep Siamese neural network based similarity model (SM) to learn the similarity between entity name variations in the KB (Fig. 1). SM consists of two identical sub-networks, that share the same weights, followed by a merge

Fig. 2. Visualization of attention learnt by Char-BiLSTM + A model. Dark color signifies more attention weight on the character.

layer that computes an energy function between them. The twin sub-networks take pre-processed sequence representations for (e_x^i, e_y^j) respectively. The output from the merge layer passes through two fully connected layers with non-linearity which finally predicts a similarity score. We will now discuss each stage in detail.

Pre-processing Entity Names: We convert all name variations e_x^i to lowercase, and remove infrequent non-informative characters (eg ‘*’). After pre-processing, the size of character vocabulary for company variations was 64. We use two types of input vectors which were fed together as input to Siamese networks: **i) Character input vector** of e_x^i is a concatenated sequence of 64-dimensional one-hot vector for each character in e_x^i . We limit the number of characters in entity name to first 43 characters with padding, which were sufficient to fully represent 95% of entities mentioned our KB. Given the character vocabulary size of 64, this resulted in each entity variation being represented by a 43 x 64 dimensional character input vector. **ii) Word input vector** of an entity name e_x^i is concatenated sequence of 100 dimensional word embeddings for every word token. Similar to i), we limit to first 7 words with padding resulting in 7 x 100 dimensional word input vector. We generate these fastText word embeddings [4,14] trained on our proprietary dataset of 50 million resumes. We follow the same pre-processing across all the entity types respectively.

Network Architecture Choices: We now describe the various architecture choices for the twin sub-networks employed in the SM. We start with describing a base network [6,19]. This architecture is incrementally improved by two additional networks described subsequently.

1. **Char-BiLSTM (Base network):** This sub-network consists of a pair of Bi-LSTM layers. The 43 x 64 dimension character vector for an entity name variation e_x^i , as described in Sect. 3.2, is fed to a pair of LSTM layers which produces a 128-dimensional feature representation h_x^i for a name variation. This architecture choice [6] forms the basis of the ECS to scale across all the entity types in our recruitment domain specific KB.
2. **Char-BiLSTM + Attention layer (Char-BiLSTM + A):** We enhance the base network by adding attention layer on top of the base network. User-generated entity variations often consists of less informative subsequences/characters. We employ attention networks [3] to generate a combined weighted representation by selectively attending to the past outputs. The weights learnt

Fig. 3. Siamese Model (SM) Architecture for Char Bi-LSTM + A+ Word + A: Character level representation learned from Bi-LSTM followed by Attention layer and word level representation learned from Bi-LSTM layer. The red vector represents the character input vector and blue vectors represents the word input vector. (Color figure online)

Table 1. Tuned hyper-parameters used for training the Siamese Model (SM). For all experiments, Batch-size is 512, BiLSTM layers are 2, Fully Connected layers in are 2, Non-linearity is ReLU and Optimizer used is Adam

Hyper-parameter	Company	Inst.	Skills	Desig.
Number of units in LSTM cell	64	64	32	32
L2 Regularizer (on LSTM weight matrix)	0.001	0.001	0.0008	0.0008
Dropout rate	0.4	0.4	0.3	0.3

by attention network allows the model to differentially attend important vs unimportant content within the sequences. Figure 2 demonstrates the role of attention mechanism while learning pairwise similarity between institute variation names. The visualization depicts how the model learns to selectively focus more certain characters in the sequence like the beginning letters, letters after a space character, and ignore less informative subsequences (eg. college, of) with respect to the entity type- institute.

3. **Combining Word and Char BiLSTM representations (Char Bi-LSTM + A + Word + A):** From our data, we observe that in the recruitment domain, many entity variations belonging to the same entity consists of synonymous subwords, grammatical variety as well as re-arrangement of subwords (eg. ‘CAD designer’ vs ‘computer-aided design draughtsman’). Character-based networks, as described above, have limited capabilities to capture similarity for such cases. Therefore, we also incorporate word-based

sequence network by introducing another pair of Bi-LSTM layers, which takes word embeddings input vector as described in Sect. 3.2 separately alongside with the Char-BiLSTM + A sub network. We concatenate the output from the Word-BiLSTM + A with the output from the the Char-BiLSTM + A, which forms the final representation h_x^i and passed to the merge layer. The final architecture **Char-BiLSTM-A + Word-BiLSTM-A** is shown in Fig. 3.

Lastly, we use a weighted merge layer (ML_1) to combine and estimate the relatedness between the entity names using the representation h_x^i and h_y^j produced by the twin sub-networks:

$$ML_1(e_x^i, e_y^j) = |h_x^i - h_y^j|$$

We use output from the ML_1 layer as input to a pair of fully connected layers that use sigmoid activation to give probability i.e. similarity score S (Fig. 3).

Model Configurations: For every entity type, we learn separate pairwise similarity models using our proposed architecture. The tuned hyper-parameters for each entity type is shown in Table 1. We use Binary Cross Entropy as the final objective function with Adam optimizer.

Table 2. Number of entity variation pairs and clusters from proprietary and public datasets.

Source	Entity type	#Variation pairs	#Entity clusters
Proprietary	Company	1237218	25602
	Institute	1196600	23690
	Skills	2909	606
	Designation	29833	3894
Public	ESCO Skills	32801	2644
	ESCO Designation	60171	2903
	DBpedia3K	182511	2949

3.3 Clustering Using Pairwise Similarity Scores

We employ Hierarchical Agglomerative Clustering (HAC), the widely preferred method [25], as our clustering algorithm to map name variations into clusters. Each cluster thus formed, would finally represent a unique real world entity. In our scenario, we use the pairwise similarity scores obtained from SM for every named-entity pair for measuring the distance/closeness between variations in HAC. We use Silhoutte index [28] to determine k , the number of clusters in HAC. This process was repeated for each dataset respectively.

4 Experimental Setup

In the experimental setup, we describe our dataset and the evaluation metrics which we use for evaluating the performance of ECS.

4.1 Dataset

We use two dataset sources (see Table 2) as described below: 1. **Proprietary:** From our Naukri database, we used 25602 company clusters, 23690 institute clusters, 606 skill clusters and 3894 designation clusters, which were manually annotated by two domain experts (with kappa agreement of 0.83). 2. **Public:** We also demonstrate the generalizability of our proposed approaches by evaluating on two open source datasets: i. DBpedia-Company: We query the large scale open KB DBpedia for extracting the type `dbo:Company` entities and their name variations. ii. ESCO-Skills and ESCO-Designation: ESCO is a recruitment domain specific open dataset. We extract the alternative labels of Skills/Competences and Occupations of ESCO to create the clusters respectively. We find that many name variations in our KB for entity type such as company and Institute were not present in Freebase (10% recall) nor in Wikipedia [9, 18] (30% recall), thus establishing that this dataset warrants a domain-specific solution.

Next, we discuss the methodology for pair generation: a) **Similar Pairs:** For every entity e_x^i , we generate its pair with every other element e_y^j such that $x=y$; if $i \neq j$ i.e. both the elements belong to the same manually annotated cluster. b) **Dissimilar Pairs:** We employ two procedures towards this task. In **Random selection** approach [19], every variation name e_x^i is paired with randomly selected variation name e_y^j from the cluster (i.e. $x \neq y$). In the second approach, we rank all the name variation of entities to create a ranked index I . Next, we query index I for every variation e_x^i and use the lower ranked variation e_y^j from results to create dissimilar name variation (e_x^i, e_y^j). We release the *similar* and *dissimilar* pairs created using our proposed method for future research. We have consolidated the entity clusters for the public dataset here².

4.2 Baselines and Evaluation Metrics

We also compare pairwise similarity from our learnt SM with the following baseline approaches: (a) **Galarraga-IDF:** When e_x and e_y have overlapping words, it may indicate they are referring to the same entity. However, not all the words might be of equal importance. Therefore, we use a weighted word overlap approach *Galarraga-IDF* proposed by [8], (b) **FastText Embeddings:** We find pairwise similarity in embeddings space of mentions as another baseline [27]. We use FastText that uses sub-word level information and robust to misspellings. (c) **Base network:** We compare our improved design choices to the base network mentioned in Sect. 3.2. We use *Precision*, *Recall* and *F1 score* [22] for evaluating the pairwise similarity results. We use *Micro-* and *Macro- Precision and Recall*, as described in ECS-1 [6] for evaluating the clusters.

² <https://github.com/anonymous-warrior/EntityCanonicalizationDataset>.

Table 3. Test Results for pairwise similarity using SM in comparison with other baseline approaches.

Method	Proprietary datasets								Open datasets							
	Company		Institute		Skill		Desig.		DBpedia-C		ESCO-Skill		ESCO-Desig.			
	P	F	P	F	P	F	P	F	P	F	P	F	P	F		
Galarraga-IDF	75.8	71.2	64.3	66.5	33.2	12.5	63	60.3	22.6	23.6	50.8	32.8	61.7	38.9		
FastText	92.9	84.6	75.8	73.1	46.7	45.2	47.9	58.5	47.5	45.3	49.7	51.6	67.9	67.1		
Char Bi-LSTM (Base)	98.4	98.7	82.5	81.8	78.8	84.7	75.3	79.2	69.5	58.8	84.9	85.9	75.2	74.1		
Char Bi-LSTM + A	99.3	98.9	84.5	84.8	81.8	86.9	72.6	77.2	72.1	59.7	85.9	86.9	76.3	75.1		
Word Bi-LSTM	95.4	95.7	79.6	83.1	79.6	86.3	89.7	93.8	77.4	70.9	85.9	86.9	76.3	75.1		
Word Bi-LSTM + A	95.3	95.6	80.6	83.3	80.1	86.5	90.5	94.8	77.6	70.7	85.6	89.6	83.1	83.7		
Char Bi-LSTM + A + Word + A	99.5	99.2	86.7	86.7	82.7	88.5	94.4	96.3	78.0	71.3	87.3	90.7	84.2	85.4		

5 Results and Discussion

Pairwise Similarity: Table 3 summarizes the evaluation of our proposed design choices Char-BiLSTM, Char-BiLSTM + A, Char-BiLSTM + A + Word + A with other baselines approaches for the measuring the pairwise similarity. We now discuss the performance of each model in detail. Table 4 discusses the qualitative analysis of the various approaches through some illustrative examples.

Galarraga IDF is a weighted word overlap based similarity measure. The main disadvantage is that it suffers from high similarity bias towards surface forms and therefore do not scale well for entities with dissimilar surface forms (e.g. ‘Cal State University Los Angeles’ vs ‘Calif. State Univ LA’). This results in poor similarity score (similarity= 0.16) even for similar variations (See Table 4) thereby resulting in poor recall (See Table 3).

Table 3 shows employing FastText increases the performance across all entities as FastText captures the semantic similarity from CV. It also helps in OOKB variations and misspellings as it generates word representations as a summation of subword representations learnt from bag-of-character n-grams. However, as shown in Table 4, using such representations for measuring similarity of an entity ‘Institute of Personnel Management Sri Lanka’ with its own variation such as ‘IPM Sri Lanka’ results in low similarity (0.54) due to lack of overlapping character n-grams. On the other hand, the summation approach results in high similarity (0.82) between dissimilar entities such as ‘Syscon Instruments Pvt Ltd’ and ‘Micron Instruments Pvt Ltd’ which share many overlapping character n-grams. Presumably, it captures limited patterns contained within sequences of characters which distinguishes one entity from the other.

Our proposed Char-BiLSTM addresses the challenge of handling complex/unseen shorthands and abbreviations which the previous models were unable to do. The sequence-to-sequence modelling *learns* these character sequence patterns from a large number of variation pairs seen in parallel by the Siamese network during training in supervised learning. The results in Table 4 show that it is infact able to output high similarity for pairs with dissimilar surface forms (‘Institute of Personnel Management Sri Lanka’ vs ‘IPM Sri Lanka’),

Table 4. Examples of pairwise similarity scores predicted using SM and other baselines.

Input pair (e_x, e_y)	True Sim.	Gal. IDF	Fast-Text	Base Sim.	Char + A + Word + A
(Cal State University Los Angeles, Calif State univ LA)	1.0	0.16	0.84	0.94	0.97
(Syscon Instruments, Micron Instruments)	0.0	0.18	0.82	0.006	0.003
(IPM Sri Lanka, Institute of Personnel Management Sri Lanka)	1.0	0.16	0.54	0.94	0.97
(London School of Design, London School of Management Education)	0.0	0.51	0.75	0.96	0.25
(fossil-fuel power plant operator, turbine operative)	1.0	0.0	0.68	0.27	0.99
(drainage technician, sewer and drain inspector)	1.0	0.08	0.73	0.17	0.99

as well as low similarity for dissimilar pairs with many overlapping characters (‘Syscon Instruments Pvt Ltd’ vs ‘Micron Instruments Pvt Ltd’). Further, these structural patterns learnt by the character model is generalizable even for out-of-KB or emerging entities.

One of the key challenges in finding pairwise similarity is the presence of synonymous substrings within the variations (eg. drainage technician vs sewer and drain inspector) (See Table 4), which leads to false negatives in character based models. For such cases, word-based sequence modelling helps in identifying the similarity between words *technician* vs *inspector*, thereby resulting in high overall similarity between the variations. Word sequence information also help in identifying dissimilarity in cases like ‘London School of Design’ vs ‘London School of Management Education’ (See Table 4). We observe from the Tables 3 that Char-BiLSTM + A + Word + A significantly outperforms all other models. A one-way repeated-measures ANOVA test showed significant difference (at 95% confidence) in all evaluation metrics (Precision, Recall, and F1-measure) across all entity types ($p < 0.0001$). Analyzing further using pairwise paired t-tests with Bonferroni correction, we validated the significance for our approach with various other baselines. Therefore, we choose Char + A + Word + A as our preferred network choice for modelling SM in ECS.

Clustering: Table 5 presents the results of applying clustering on pairwise similarity to identify unique entity groups used to canonicalize name variations in the test dataset. We find that our approach consistently performs well across all the entity types. In the proprietary dataset, we achieve best performance for company entity with 0.98 F1-scores. Table 6 demonstrates the efficacy of our approach in canonicalizing non-trivial instances involving (dis)similar surface forms and emerging entities. For example, in case of company entity type, ECS identifies various canonical forms for the entity ‘Warner Brothers Studios’ despite the existence of a variety of dissimilar surface forms like ‘warnerbros.com’, ‘warner brothers international television’ and ‘warner bros. entertainment’. We believe that while Char-BiLSTM could successfully capture the variety of dissimilar

Table 5. Test Results for clustering using HAC over pairwise distances predicted by SM.

Model	Micro			Macro		
	<i>P</i>	<i>R</i>	<i>F1</i>	<i>P</i>	<i>R</i>	<i>F1</i>
Company	0.98	0.99	0.98	0.97	0.96	0.96
Institute	0.84	0.75	0.79	0.96	0.48	0.64
Designation	0.95	0.53	0.67	0.83	0.15	0.24
Skill	0.71	0.64	0.67	0.94	0.31	0.47
DBpedia-Company	0.88	0.52	0.65	0.92	0.25	0.39
ESCO-Designation	0.49	0.79	0.61	0.21	0.32	0.25
ESCO-Skill	0.84	0.82	0.83	0.65	0.49	0.55

Table 6. Sample ECS entity clusters generated from our proposed ECS.

Entity type	Entity name	Predicted entity clusters	Incorrect variation(s)	comments
Comp.	<i>Warner Brothers Studios</i>	['warner brothers pictures', 'warner bros.-first national', 'warnerbros.com', 'warner brothers studios', 'warner bros.', 'warner bros. entertainment', 'warner bros', 'warnerbrothers films', 'warner bros. entertainment inc.', 'the warner brothers studio']	None	ECS is correctly able to include complex variations like 'warnerbros.com'. Also, similar word suffixes like "entertainment", "film" and "studios" help in improving the recall
Skills	<i>Floor Management</i>	['Floor managing', 'Floor management', 'Floor Manager', 'Floor Planning']	Floor Planning	Incorrect inclusion due to same prefix "floor" and similar embeddings for words "managing" and "planning"
Skills	<i>quality assurance procedures</i>	['quality assurance procedures', 'quality check', 'quality control procedures', 'inspection procedures', 'QA procedure']	None	ECS is able to identify semantic relatedness like "inspection" and "quality assurance", and shorthands like "QA"
Desig.	<i>Fishery Production Manager</i>	['Fish Farm Production manager', 'Fishery production Manager', 'Aquaculture Production manager', 'Manager of Aquaculture Production', 'Horticulture Manager', 'Director of Horticulture']	'Horticulture Manager', 'Director of Horticulture'	Merged with the cluster of 'horticulture manager' which had semantically similar sequences related to food production/farming

surface forms such as shorthands and spelling variations, the attention mechanism could help to focus on the important terms like *warner* while disregarding noisy but lengthy strings like 'first national'. Moreover, the word embeddings could help in boosting the similarity due to the presence of synonymous terms like *entertainment*, *film* and *studios*. We find similar trends in other entity types.

Error Analysis: Table 6 demonstrates some of errors encountered. Sometimes individual word sequences are unable to capture the meaning of the entire phrase. For example, the skill variations ‘Floor management’ and ‘Floor planning’ are very different activities. However, due to the overlapping of word *floor* and presence of similar words like *management* and *planning*, they incorrectly get mapped into same cluster.

6 Conclusion

Our research focuses on developing a novel solution for entity canonicalization in KBs for the recruitment domain. We canonicalize entities like institutes, companies, skill sets, and designation because: a) of their importance to the recruitment domain for various applications such as recommendation and search; and b) as these entities have the most variations in their representations in KBs for recruitment. The proposed Similarity Model (SM) learns the entity pair similarity using Siamese neural networks comprising of Bi-Directional LSTMs with attention on character embeddings. Using word embeddings, we enhanced the capability of the model to capture the semantic similarity. Further, we applied Hierarchical Agglomerative Clustering (HAC) using these pairwise similarities to create clusters of unique entities. In our work, we efficacy of our architecture on both proprietary and public dataset. Experiments revealed that the Siamese neural network-based approach learns a character level relationships and achieves competitive performance relative to the baselines. We believe our proposed methods are also generalizable to entities from other similar domains that lack sufficient contextual information in open KBs such as Wikipedia.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC -2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76298-0_52
2. Bagga, A., Baldwin, B.: Entity-based cross-document coreferencing using vector space model. In: Proceedings of the 36th Annual Meeting of ACL (1998)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
4. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. *Trans. ACL* **5**, 135–146 (2017)
5. Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T.: Entity disambiguation for knowledge base population. In: Proceedings of the 23rd International Conference on ACL (2010)
6. Fatma, N., Sachdeva, N., Rajput, N.: Canonicalizing organization names for recruitment domain. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (2020)
7. Francis-Landau, M., Durrett, G., Klein, D.: Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. [arXiv:1604.00734](https://arxiv.org/abs/1604.00734) (2016)

8. Galárraga, L., Heitz, G., Murphy, K., Suchanek, F.M.: Canonicalizing open knowledge bases. In: Proceedings of the 23rd International Conference of CIKM (2014)
9. Goldsmith, J.: Wikipedia API for Python. <https://pypi.org/project/wikipedia/>
10. Gupta, N., Singh, S., Roth, D.: Entity linking via joint encoding of types, descriptions, and context. In: Proceedings of the EMNLP (2017)
11. He, Z., Liu, S., Li, M., Zhou, M., Zhang, L., Wang, H.: Learning entity representation for entity disambiguation. In: Proceedings of the 51st ACL (2013)
12. Hoffart, J., Altun, Y., Weikum, G.: Discovering emerging entities with ambiguous names. In: Proceedings of the 23rd International conference on WWW (2014)
13. Hoffart, J., et al.: Robust disambiguation of named entities in text. In: Proceedings of the Conference on EMNLP (2011)
14. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the EACL, April 2017
15. Kardes, H., Konidena, D., Agrawal, S., Huff, M., Sun, A.: Graph-based approaches for organization entity resolution in mapreduce. In: Proceedings of TextGraphs-8 Graph-based Methods for Natural Language Processing (2013)
16. Lin, T., Etzioni, O., et al.: Entity linking at web scale. In: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction. ACL (2012)
17. Liu, Q., Javed, F., Mcnair, M.: CompanyDepot: employer name normalization in online recruitment industry. In: Proceedings of the 22nd ACM SIGKDD (2016)
18. MediaWiki: Api:main page – mediawiki, the free wiki engine (2018). https://www.mediawiki.org/w/index.php?title=API:Main_page&oldid=2988334
19. Neculoiu, P., Versteegh, M., Rotaru, M.: Learning text similarity with siamese recurrent networks. In: Proceedings of the 1st Workshop on Representation Learning for NLP (2016)
20. Pakhira, M.K.: A linear time-complexity k-means algorithm using cluster shifting. In: International Conference on Computational Intelligence and Communication Networks (CICN). IEEE (2014)
21. Pujara, J., Miao, H., Getoor, L., Cohen, W.: Knowledge graph identification. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8218, pp. 542–557. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41335-3_34
22. Raghavan, V., Bollmann, P., Jung, G.S.: A critical investigation of recall and precision as measures of retrieval system performance. ACM Trans. Inf. Syst. (TOIS) **7**, 205–229 (1989)
23. Shen, W., Wang, J., Han, J.: Entity linking with a knowledge base: issues, techniques, and solutions. IEEE Trans. Knowl. Data Eng. **27**, 443–460 (2015)
24. Sun, Y., Lin, L., Tang, D., Yang, N., Ji, Z., Wang, X.: Modeling mention, context and entity with neural networks for entity disambiguation. In: Twenty-fourth International Joint Conference on Artificial Intelligence IJCAI (2015)
25. Tan, P.N., et al.: Introduction to Data Mining. Pearson Education India, Bangalore (2006)
26. Trani, S., Ceccarelli, D., Lucchese, C., Orlando, S., Perego, R.: Dexter 2.0: an open source tool for semantically enriching data. In: Proceedings of the 2014 International Conference on Posters & Demonstrations Track (2014)
27. Vashishth, S., Jain, P., Talukdar, P.: CESI: canonicalizing open knowledge bases using embeddings and side information. In: Proceedings of the WWW conference (2018)

28. Wikipedia: Silhouette (clustering) – Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=Silhouette_\(clustering\)&oldid=856197782](https://en.wikipedia.org/w/index.php?title=Silhouette_(clustering)&oldid=856197782) (2018)
29. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: ACM Sigmod Record (1996)
30. Zhang, W., Su, J., Tan, C.L., Wang, W.T.: Entity linking leveraging: automatically generated annotation. In: Proceedings of the 23rd International Conference on ACL (2010)