International Journal of Computational Intelligence Systems
Vol. **13(1)**, 2020, *pp.* 1002–1013
DOI: https://doi.org/10.2991/ijcis.d.200706.002; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Coreference Resolution Using Semantic Features and Fully Connected Neural Network in the Persian Language

Hossein Sahlani[1,*], Maryam Hourali[1], Behrouz Minaei-Bidgoli[2]

[1]*Malek-Ashtar University of Technology, Tehran, Iran*
[2]*School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran*

ARTICLE INFO

Article History
Received 09 Feb 2020
Accepted 05 Jun 2020

Keywords
Coreference resolution
Fully connected neural networks
Deep learning
Hierarchical accumulative clustering
Persian language

ABSTRACT

Coreference resolution is one of the most critical issues in various applications of natural language processing, such as machine translation, sentiment analysis, summarization, etc. In the process of coreference resolution, in this paper, a fully connected neural network approach has been adopted to enhance the performance of feature extraction whilst also facilitating the mention pair classification process for coreference resolution in the Persian language. For this purpose, first, we focus on the feature extraction phase by fusing some handcrafted features, word embedding features and semantic features. Then, a fully connected deep neural network is utilized to determine the probability of the validity of the mention pairs. After that, the numeric output of the last layer of the utilized neural network is considered as the feature vector of the valid mention pairs. Finally, the coreference mention pairs are specified by utilizing a hierarchical accumulative clustering method. The proposed method's evaluation on the Uppsala dataset demonstrates a meaningful improvement, as indicated by the F-score 64.54%, in comparison to state-of-the-art methods.

1. INTRODUCTION

The progress of information and machining work, show that information extraction and the use of text analysis systems are an essential problem. However, there might be ambiguities in these systems that can be resolved by different algorithms.

One of the ambiguity in natural language processing is incorrect coreference resolution, in general, to avoid repetition when expressing more information about an entity, do not use the full name of it, and use different descriptions, e.g., use a pronoun or his first name. These terms, which are used to refer to an entity, are called "mentions," and all mentions refer to the same entity, are coreference together, which these are noncoreference with other mentions.

Coreference resolution is the task of finding all expressions that refer to the same entity in a text. For example, there are many mentions in the following text snippet from which more important ones are marked with brackets.

[Mexican football]$_{m1}$ got [a boost]$_{m2}$ in [September]$_{m3}$ when [former Brazil and Barcelona star]$_{m4}$ [Ronaldinho]$_{m5}$, joined [modest local club [Queretaro]$_{m6}$]$_{m7}$. [Carlos Trevino]$_{m8}$, [a former official of the [Queretaro state government]$_{m9}$]$_{m10}$, launched [an attack]$_{m11}$ on [Ronaldinho]$_{m12}$ before [the Brazilian]$_{m13}$ had played [a single game]$_{m14}$.

In the above text mentions {m4, m5, m12, m13}, {m8, m10} are coreference together.

Coreference resolution is one of the most challenging issues in the field of natural language processing [1]. The coreference resolution is an essential key in text comprehension, and have significant application in natural language processing, e.g., information extraction, summarization, questions answering, machine translation, etc. [2–5].

Coreference resolution is one of the challenges issues of natural language processing not only in the English language but also in all of the languages. To solve these challenge are two solutions: the first is language-dependent, which these methods solve coreference resolution according to the signs that exist in some languages, and machine or algorithm can be more easily solve this challenge, such as gender matching in the Arabic language. The second solution is language-independent, which these methods, solve the Coreference resolution and offer solutions that can be used in all languages. In the related works to solving the coreference resolution, both of the language-dependent and language-independent approaches are used.

In general, coreference resolution methods divided into two main categories: machine learning-based and rule-based methods [6]. In the rule-based methods, a set of rules are written to determine the coreference resolution. The advantage of this method is its simple design. However, the flexibility of this method is low, so

* *Corresponding author. Email: sahlani@mut.ac.ir; sahlani_h@yahoo.com*

all languages must be redesigned from the beginning by experts [7]. Machine learning methods are divided into supervised and unsupervised classes. In supervised strategies, existing training data needs to be annotated by an expert [3].

In the following sections of this paper, the related work done to the Persian and English language coreference resolution are explained; then, are expressed how to recognize name entities and how to extract features from mention pairs and how to train our model. In feature extraction, are used three categories of features to extract essential features; handcrafted features [8], word embedding features, and semantic features are used, and in the training model phase, a fully connected neural network and a hierarchical clustering method are used. Next, the proposed method evaluated in the Uppsala database [9]. Then the results of the proposed method are described and compared with other related methods, that show improvement by approximately 5% compared to the method in [8] and reach 64.54% accuracy and finally, after analyzing the error in the experiments, we state the conclusion and summarize the paper.

2. RELATED WORKS

The coreference resolution is divided into two categories: the entity coreference and the events coreference, in which the principles are the same, and one can use the ideas of others. The related works in the events coreference are Lu work's [3], which first, they determined essential features and reduced the complexity by removing other features, then used Markov Logic Networks to coreference resolution. The same authors in [6] presented a coreference resolution using a binary classification on the KBP 2016 corpus. In [10], first, the authors expressed the challenges in the event coreference; then, they resolve coreference by extracting important features of mentions. Authors in [11], try to resolved unsupervised events coreference. In [12], the authors considered two distance criteria for in-text and out-text features and combined the value of them to increase the accuracy of the event coreference resolution.

This paper proposed an entity coreference resolution, which works on the entity coreference considered in two categories: rules-based methods and machine learning-based methods (Figure 1).

Coreference resolution methods can express in three main steps:

The first step (corpus creation and mention pairs detection), corpus creation is a work that requires time and expertise in this field. There are also many suggestions about how to code corpus, and each corpus uses its encoding, and there is no general standard for it, if there is an appropriate corpus, it can skip this part. CoNLL [13] is a proper corpus for coreference resolution in some languages, such as English, Chinese and Arabic. Authors in [8] created a Persian corpus according to the CoNLL standard.

After selection or creation of corpus need to detect mention pairs, in some methods, mention pairs identified with preprocessing such as parts of speech, parser, etc., and in some of the other methods use the information of standard golden corpus such as [8,14,15], etc.

Coreference resolution algorithms usually detect mention pairs in three different methods: the first is the mention pair model (the mention pairs has classified as a coreference or as noncoreference), the second is the entity-mention model (classify each mention with a previous entity) and the third is the ranking model (calculate the possibility of the mention pairs and sort them and select the maximum coreference possibility) [3]. In the Table 1 seen the used method for the detection of mention pairs.

In the second step (feature extraction or applying rules), extract the features from the mention pairs. In machine learning-based, various features are used to determine coreference resolution, such as word embedding vector in [1,14,15], etc., distance, head matching words, etc., in [1,8,14,15] and, etc., which are directly extracted from corpus, semantic features, etc.

In rules-based methods, applied important rules from top to down to detect the mention pairs and eliminate unlikely and incompatible mentions, it should note rules are language-dependent [7,16]. Figure 2 shows rule-based flowchart methods that state the rule importance from top to down.

Due to the complexity of the coreference, the rules-based systems transformed into machine learning-based systems; there are several methods for extracting the appropriate features and giving them appropriate weight. The machine learning-based algorithms lead to better sorting and better weighting of features in comparison to the rules-based systems.

Extraction of the appropriate features is difficult that researchers did different act to extract these features. Some of them extract features that directly exist in the text, which named these features handcrafted, that in different languages, different actions have been taken in this field, most of these are language-dependent, such as gender agreement features, headword matching, plural or singular agreement, etc.

Some researchers try to extract features which not directly in the text; these features are understandable to humans inadvertently and do not exist directly in the text and must be learned and extracted by the machine. For this purpose, need to external resources, which one of the best external resources is the web. To extract information from the web, must be implemented a corpus, which can detect communication between words and determine whether there is a connection between these words or not? These features named

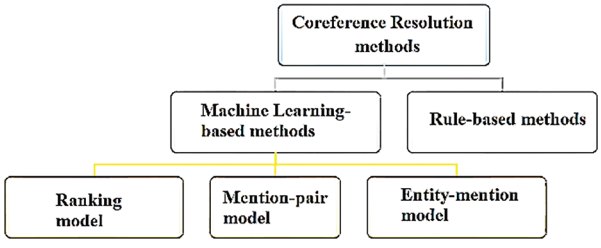

Figure 1 Coreference resolution methods.

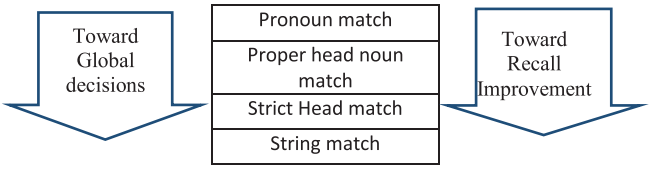

Figure 2 The priority of some rules used in the rule-based system.

semantic in this paper. For this purpose, researchers have been created different corpora like Yago [17], DBpedia [18], WordNet [19], etc., in the English language and FarsBase [20] in the Persian language.

Authors in [21] investigated two publicly available web knowledge bases, Wikipedia and Yago [17], in an attempt to leverage semantic information and increase the performance level of a state-of-the-art coreference resolution engine. They extracted semantic compatibility and aliasing information from Wikipedia and Yago [17], and incorporate it into a coreference resolution system.

They in [21] defined two features, Yago-means and Yago-type as follows:

The Yago-means feature set to true if the Yago entries stay in means relation (we add reflexive means relations from each node to itself).

The Yago-type feature set to true if (a) there exist Yago nodes T_1 and T_2 such as the anaphor stay in type relation with T_1 and (b) the antecedent stay in means relation with T_2 and (c) T_1 and T_2 stay in type relation or vice versa.

Another way to extract the high-level features, which are understandable to humans, is to use neural network systems to learn understandable concepts to the machine, such as dictionary word algorithm, word2vec [22] algorithm, Glove algorithm [23], etc. These features named word embedding in this paper too.

In the third step (training model), the proposed model has been trained by extracting features from the previous step. There are several algorithms to train the model, according to the various available algorithms for this purpose, authors in [8] compared several of these algorithms, such as decision tree, naive Bayesian and the SVM, which showed the SVM is better than others. As well as authors in [14,15] used the Fully Connected neural network with a large number of nodes (which name this method is deep learning), which showed this classification has high accuracy. Authors in [1] used Bultzman deep neural network for coreference resolution, which has high accuracy. They show that the fully connected neural network is better than other classifications methods.

By considering that, Coreference Resolution is a binary classification and the output of this classification is 0 or 1, some of the methods use a clustering algorithm to apply coreference mention pairs in the same cluster such as [14].

In the Table 1, coreference resolution systems listed in different categories.

As can be seen, Table 1 stated the classification of the related works. It can consider several categories for some methods, e.g., in the [29,30] and [6], used the entity mention and ranking model; or in [14,15], used deep learning methods to classify mention pairs and a clustering algorithm to specify coreference mentions in the same cluster. These methods used entity mention method and ranking model.

In addition to the category of methods seen in the Table 1, we used the features stated in these papers, such as the matching headword, distance, etc. It should be noted that many features are language-dependent and unable to use in all languages. Also, some features need preprocessing such as databases, e.g., in [32] and [21] used the Yago [17], DBpedia [18], or other resources that only exist in the English language.

3. PROPOSED METHOD

In this section, an approach for coreference resolution based on graph and fully connected neural networks presented. In each document, features extraction are performed after finishing the preprocessing phase and mentions detection. In the proposed method to extract features of the mention three categories features are used: the first, handcrafted features (essential features in different papers), the second, the word embedding vectors and the third, semantic features which extracted from Wikipedia. The extracted features from the mentions are used as input to the fully connected neural network to train the model, after training of the model, detected the probability of coreference mention pairs.

The coreference mention pairs used as input to the graph, to cluster similar coreference mention pairs in one group. In other words, after clustering of the graph, the similar mention pairs are placed in a group which is known as the coreference. Figure 3 shows the steps of the proposed algorithm.

It should note that the similarity of nodes determined by features that extracted from the mention pairs in the training step, these features are the probability of coreference mention pairs and the new features that created before the determination of the likelihood of coreference mention pairs.

The proposed method divided into four general sections:

The first step (Corpus Creation and Preprocessing): Access to the desired corpus in some languages is difficult, and sometimes the researcher must first create the corpus. Authors in [8] created coreference resolution corpus in the Persian language, and we used this corpus in the proposed method. After corpus creation, name entities should be recognized in input texts. This step depends on a corpus, in some corpora entities named as golden data, we used corpus in [8] which determined name entities itself.

The second step (Feature Extraction): In this step, after recognizing the named entities, the mention pairs are determined, and features from them are extracted. The proposed method uses three categories of the feature: the first is word embedding vector, the second is semantic features extracted from Wikipedia and the third is handcrafted features, which are the distance between the mentions, head matching, gender matching, etc.

The third step (Probability Assignment or Mention Pairs): A Feedforward Neural Network (FFNN) is trained by the candidate mention pairs (extracted features from them) and their labels (presence/absence or 1/0) so that the trained FFNN assigns a probability (between 0 and 1) to any given mention pair.

The fourth step (Coreference Graph Resolution): This step creates the graph by using the extracted mention pairs from the previous step. In this graph, nodes are the mention pairs that are clustered by using the agglomerative hierarchical clustering algorithm to locate similar mention pairs in a group. The resulting clusters are considered as coreference resolution chains

Table 1 Some of the critical coreference resolution method.

Method	Method Idea	Models type	Learning type
[24]	Coreference resolution in name entities such as pronouns, place nouns, etc.	Mention-pair	Supervised
[3]	This method selected important features, reduced dimensions, and coreference event resolution with Markov logic networks.		
[25]	In this method are used Wikipedia and coreference resolution to information extraction and questions answering.		
[21]	Using YAGO, Wikipedia, etc. to semantic features extraction and eliminating ambiguous information then create a taxonomy and compared it with other resources such as WordNet, etc.		
[14]	Using the word embedding and handcrafted features, and a fully connected neural network and reinforcement learning method to train the model.		
[15]	As same as previous paper [14], after extracting features and training the model, specification of coreference mentions in a cluster is needed.		
[26]	This article study entities and event coreference resolution		
[10]	Investigating the challenges and using important features in events coreference resolution		
[27]	Using clustering and classification then prune the graph edge (BESTCUT)	mention-ranking	
[28]	Use of ranking models instead of classification model in entities coreference resolution		
[29]	Representing a hidden Markov model for coreference resolution	entity-mention	Unsupervised
[30]	Using a distance hierarchical Bayesian model for unsupervised events coreference resolution		
[16]	This paper makes coreference resolution by using a hidden Markov model and compares this with(Haghighi & Klein, 2009).		
[31]	Specify coreference mentions in a cluster		
[6]	Presenting events coreference resolution by ranking training model and use of related features with using of binary classification on KBP 2016corpus		Supervised
[32]	Using YAGO, FrameNet, etc. for semantic features extracting		
[11]	Using mentions features, mention pair features, and clustering to coreference resolution		
[12]	Events coreference resolution considering measuring criteria of distance mentions in textual and contextual and combining them		
[33]	Presenting the CEAF model to solve the problem of assessing criteria MUC and B3		
[1]	Used deep network LSTM, CNN, to detect mention pairs and used word embedding and handcrafted features to coreference resolution.		
[7]	Causing a high level of accuracy with syntactic and semantic features		

Figure 3 represents a schematic view of the block diagram of the proposed method. The following subsections explain the proposed method more in detail.

3.1. Corpus Selection and Preprocessing

In the coreference resolution task, the results of the machine learning methods have been better than the rule-based methods. Machine learning methods need to have an appropriate corpus that has been tagged by experts. Different appropriate coreference resolution corpora have been created in the English language; however, in the Persian language, it has been constructed into an appropriate corpus by [8], which is comparable with appropriate corpora such as the CoNLL. This corpus is composed of news put out by the most popular news agencies in Iran during September 2016. It includes subjects such as economic news, technology, politics, sport, society, culture and art news. This corpus has more than one million Persian words with a gold coreference label, which means the coreference has been tagged manually and automatically.

RCDAT Corpus [8] is similar to the CoNLL corpus and contains 12 fields for any mention. All fields of this corpus include the following:

``File Name or Document ID, Part number, Word itself, Part-of-Speech -16 (identifies POS with 16 labels), Named Entities labels_13 (with 13 Gold labels), Word tokens, predicated lemma, Named Entities labels_3 (with 3 Golden labels), Index of the word in Coreference chain (Golden label), Coreference chain number (Golden label), Type of Mention, gender, Type of phrase, POS_100."

After corpus selection, the first step is the detection of mention pairs; in some of the corpora, the mentions have been detected and can skip this step, e.g., the CoNLL corpus in English or the corpus in [8].

After corpus selection, the first step in coreference resolution is name entity recognition and mention pairs detection. There are some problems in recognizing the named entities in the corpus, i.e., mismatching of recognized named entities and gold-named entities, or sometimes, the recognized named entities are not in the golden corpus, or the named entities cannot be recognized in the text [6]. To reduce any error in coreference resolution, the authors of [8] had to do a series of preprocessing operations on the corpus text.

In the preprocessing step before coreference resolution, the corpora are normalized, checked using spell correction and sentences split,

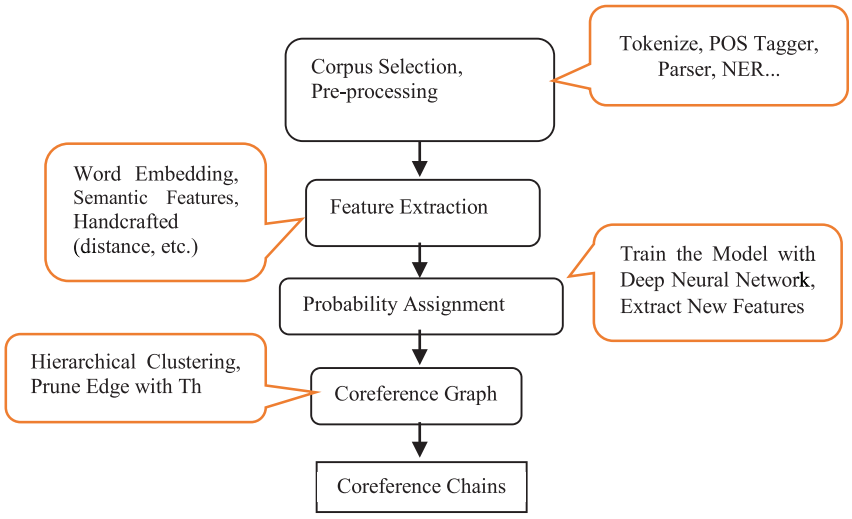

Figure 3 The block diagram of the proposed method.

Table 2 Accuracy of pre-processing tools in the RCDAT corpus [8].

Tools	Accuracy (%)
POS	98
NP-chunker	70
NER	85
Paragraph splitter	98
Sentence splitter	98
Tokenizer and spellchecker	93

etc. The accuracy of each of the preprocessing tools used in [8] is presented in Table 2.

Table 2 shows the accuracy of preprocessing methods, the low accuracy of which reduces the accuracy of the proposed coreference resolution method.

3.2. Feature Extraction

After name entities recognition phase, mention pair together, it is necessary to extract mention pairs features to train these. These features divide into three-part: the first part is word embedding features, the second part is handcrafted features and the third part is semantic features.

3.2.1. Word embedding features

Neural networks are used widely in various machine learning fields to combine features in higher layers and create new features. A fully connected neural network is a collection of neural network models that can be taught nonlinear transformations of data and attempt these to high-level concepts. Deep neural networks are developed from neural networks to extract concepts at different layers.

The deep learning algorithm is used in natural language processing to provide an in-depth analysis of the text; i.e., it extracts important features from mention pairs in coreference resolution. Deep fully connected models are used to gain advanced features from the mention pairs. The results show that extracting features from fully connected models increased accuracy. One approach for using words and sentences in a variety of machine learning numerical methods, such as classification algorithms, is a numerical representation of words and sentences.

One approach is to use an n-word dictionary that represents each word in a vector with dictionary length so that all of the nodes are zero except the corresponding node to that word. Sentence vector can also be represented by a vector, which is a combination of the word vector used with it. Even though it is simple, it requires a large amount of memory space, and in this vector, only words and their repetition are essential, so the order of the words or the subject of the document does not affect the model.

Another of the methods used in this area is the Google Word2Vec algorithm [5], which is an efficient way to display words and documents. In this method, a unique representation vector is given for each word. After creating vectors related to each word, vectors can find the mean of each column's numbered vector of words that are used within it to display each sentence (Figure 4).

Another method for the numerical representation of words is Glove algorithm [23], which is superior to the previous methods. Glove is an efficient method for displaying words as vectors. In this method, a unique fixed-size vector with a deep neural network is created for each word, and an average vector of words in it is considered for each sentence.

This paper uses the Persian Wikipedia corpus to vector the words. The results of the implementation of the Glove algorithm [23] on this corpus are unique vectors with a length of 300, which this length can change [34].

In this paper, the embedding method has done in two ways: the first is to obtain the word embedding vector for each mention pair and comparing the similarity of them. The second is to examine the similarity of mention sentence embedding with the candidate entity.

For example, in "[Tehran city] is [Qom city's] neighbor and [Massoumeh Shrine] is in [this city]." The similarity of the [Qom] and the sentence "[Massoumeh Shrine] is in [this city]" is more similar than [Tehran city] and this sentence; therefore, [this city] and [Qom] are coreference.

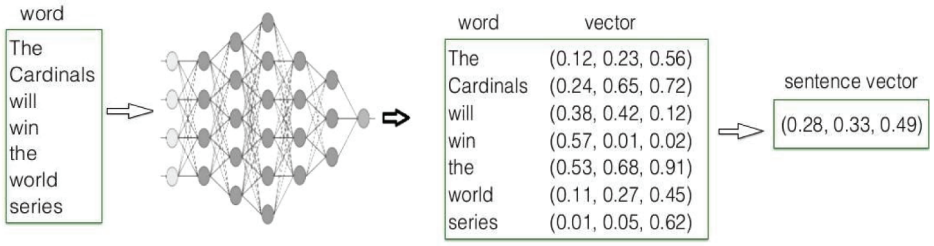

Figure 4 An example of text to vector converter.

3.2.2. Handcrafted features

In supervised systems, mention pair samples ($\mathbf{m_i}$, $\mathbf{m_j}$) are labeled true (coreference) or false (noncoreference).In this paper, $\mathbf{m_i}$, $\mathbf{m_j}$ are mention pairs, and their **label** is true if $\mathbf{m_i}$ and $\mathbf{m_j}$ are coreferenced together. A features vector represents the mention pairs of $\mathbf{m_i}$ & $\mathbf{m_j}$; this can be used with these labels as criteria for evaluating methods. Traditional methods of coreference resolution used handcrafted features, which mainly depend on the prior knowledge of designers, and some of the handcrafted features are dependent on language or some of the preprocessing in language. The handcrafted features that have been used in our proposed method are

``the first mention, the second mention, the distance between the two mentions (in the sentence), the type of the mentions, the mentions string matching(first name matching, the head matching), the existence of definitions word in the mentions, Speaker detection, number agreement (singular or plural), gender agreement, the living mention matching."

In the following, we point out some of the challenges regarding handcrafted features and detail of some handcrafted features which are used in this paper:

- *Distance Features*: One of the essential features in coreference resolution is the distance of the mention pairs in a document, and this can be used inside of other features; i.e., string matching features of mention pairs in two sentences can be coreference regardless of the distance between them, but gender agreement features between a pronoun and mention can be useful only in a similar sentence or at least once in a sentence space. Therefore, the distance features can be a supplement for many other features [7]. So selecting coreference candidates will usually be done concerning their distance because it is assumed that the boundaries of an entity are near to the entity. Distance features can be considered as words, phrases, paragraphs, or the whole document. In this paper is assumed three sentences for the boundaries of coreference resolution and distance feature is considered as words, sentences and paragraphs.
- *String Matching Features*: These features include head matching, exact string matching and partial string matching.
- *Gender Agreement Features*: Any coreference mentions should agree on their gender, i.e., male, female, both or nonliving entity. Gender is an essential feature in anaphora resolution [5], which prunes the candidate mentions' search space, i.e., in the sentence "*Ali (1) bought a book (2). He (3) is clever*," (2) and (3), is disagreement, (1) and (3), is an agreement in gender (*He* = *Ali*). Mention gender agreement is y/n if they are/are not in agreement or u if at least one of the mentions have no apparent gender. It should be noted that the gender agreement features in the Persian language is less critical because there are no separate male or female pronouns.
- *Number Agreement Features*: An entity is a coreference with its mentions if they agree on singularity or plurality. For example, in the sentence "*Fatima and her sisters (1) go to the university (2) by taxi (3), and they (4) met Sara (5) in there (6).*" The pronominal reference (4) refers to (1) Referent, (2) and (3) are successfully eliminated based on number disagreement. The number agreement and gender agreement features in Arabic and English languages are more important than in the Persian language because in the Persian language plural pronouns are usually used to show respect to a person instead of singular pronouns and pronouns in the Persian language never state the gender of the references, i.e., in "*he/she comes*," this may be referred to "*Ali or Fatima*" in the Persian language.

For example, in "*Maryam (1) did not go to her (2) school because she (3) was sick)*," the handcrafted features of "*(1), (3)*" in the Persian language are as follows:

Maryam, she, 0, 7, 0, 0, 1, 0, 0, 0, 0, 0, 1, 2, 1.

(first mention(Maryam), second mention(she), distance (0,7,0), type of the mentions(0, 1), string matching(0,0,0), definitions word (0), Same Speaker(0), number agreement (1), gender agreement(2 unknown), living agreement (1)).

3.2.3. Semantic features

In the last decade, some of the researchers try to extract rich features which not directly in the text, should be noted in the preview, these features are understandable to humans inadvertently and do not exist directly in the text.

These features in this paper named semantic.

For this purpose, need to external knowledge, which one of the best them is information on the web. To extract information from the web must be used from the semi-structured corpus, which can detect communication between words, such as Wikipedia but this corpus not relational, so some of the extensive research has been done on constructing knowledge graphs. Ref 20 includes knowledge graphs constructed from Wikipedia such as DBpedia [18]; systems that extract knowledge raw text, e.g., NELL; as well as the hybrid systems that exploit multiple types of information sources, including Yago [17]. Construction a multi-domain and comprehensive

knowledge graph from unstructured and semi-structured web contents have been of interest for a while. The DBpedia project was initiated a decade ago to construct a knowledge graph from Wikipedia. Further works like Yago, etc., integrated other sources, e.g., WordNet and GeoNames, in order to construct a more enriched knowledge graph. However, all of them are in the English language, so authors of [20] present FarsBase, a Persian knowledge graph constructed from various information sources, including Wikipedia, web tables and raw text. FarsBase is specifically designed to fit the requirements of structural query answering in Persian search engines.

In some past related works, such as [32] and [21], to improve the accuracy of the coreference resolution system, considered the relation of the mention pairs in existing knowledge databases such as Yago [17].

YAGO's unification of the information in Wikipedia and WordNet enables it to extract facts that cannot be extracted with Wikipedia or WordNet alone, such as (Tehran, TYPE, City).

Yago is an extracted ontology from Wikipedia and merged with WordNet which examined semantic relationships between the mention pairs; thus Yago system returns value "1" when both of the mentions are related, e.g. ["New York," "city"] and "0" when there is no relationship between them, e.g. ["Obama," "city"].

Authors in [32] incorporated the world knowledge from YAGO into our coreference resolution models as a binary-valued feature. If the mention pairs model is used, the YAGO feature, for instance, will have the value "one or 1" if and only if the two noun phrases involved are in a TYPE or MEANS relation. On the other hand, if the coreference resolution model is used, the YAGO feature, for instance, involving noun phrase NP*k* and preceding cluster c will have the value "one or 1" if and only if NP*k* has a TYPE or MEANS relation with any of the noun phrases in *c*.

In the proposed method, this feature is created using the FarsBase Knowledge Base [20]. This database has collected redirects in Persian Wikipedia. For example, in

``[Dr. Hassan Rouhani] [Islamic Republic of Iran President] was officially welcomed by his Turkish counterpart Rajab Tayyip Erdogan on Thursday at the Turkish Presidential Palace."

The phrase of [Dr. Hassan Rouhani] and [Islamic Republic of Iran President] have been considered in the knowledge base as a related.

To improve accuracy, in the proposed method, firstly removed the stop words in the mention pairs then find the relation between of them on the FarsBase Knowledge Base [20]. The relation showed with binary value.

3.2.4. Features concatenation

In this step, the extracted features are concatenated with other features and are used as an input vector for the training proposed method phase. Extracted features have divided into three categories:

- Word embedding features: mention pair similarity and the similarity of each mention with another mention sentence.
- Handcrafted features: mention pair distance (by sentence), strings matching, gender matching, number matching, etc.
- Semantic features: relation of mention pair in the FaresBase knowledge base [20].

For example, in "*Ali (1) did not go to his office yesterday because he had a meeting with Mohsen (2)*," we want to detect whether "*(1), (2)*" are coreference with each other or not. The concatenated feature of these are as follows:

Ali, Mohsen, handcrafted features {0, 14, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1}, word embedding features {similarity of mention pairs (0.74), similarity of mention and mention sentence (0.78)}, semantic features {0}.

3.3. Probability Assignment: Mention Pairs

In the training phase of the proposed method, a fully connected neural network is used, which firstly determines mention pairs that are labeled in the RCDAT corpus [8]. Then the features extracted in the previous steps are given as input data to the neural network and so the model becomes trained. Figure 5 shows the fully connected neural network architecture of the proposed method.

As seen in Figure 5, the neural network architecture of the proposed method has three parts, which are described below:

Input layer: An input vector (w) is considered for each of the mention pairs. The input vector contains three categories of features: words embedding features, semantic features and handcrafted features.

Hidden layers: The feature vector is transferred as input to the first hidden layer. The number of hidden layers is three, each hidden layer is fully connected to the previous layer, and the function used in these hidden layers is ReLU.[1] The number of hidden layered neurons are M1 = 800, M2 = M3 = 400.

Output layer: The output of the last hidden layer is transferred to the output layer, and the function used in the output layer is sigmoid; thus, the output of the neural network is a probability and a real number between 0 and 1. In addition to the final output of the neural network, the output of the third hidden layer is used in the graph formation phase.

3.4. Coreference Graph Resolution

In the training phase, only local information between mention pairs is considered. We use global information in training a graph-based model that clusters the mention pairs.

In this phase (formation graph), it is assumed that all the coreference mention pairs from the previous step are graph nodes and that all nodes are interconnected nondirectionally together, and all edge weights are initially zero. At the starting state, each mention pair is specified as cluster G_i then assigned a score $S_{gh}(G_1, G_2)$that represents the compatibility of the cluster G_1, G_2, and shows the probability of coreferencing G_1, G_2. Clustering does this by using a hierarchical clustering algorithm and resolves all the coreference mention pairs. After agglomerative hierarchical clustering, these weights are changed, and all coreference mentions are placed into the same clusters. The weight of the edges is calculated by using various features that have been mentioned in the previous sections.

[1]Rectified Linear Units.

These features are extracted from the last hidden layer in the previous step and divided into two groups, such as increasing and decreasing edge weight.

Figure 6 shows the coreference mentions clustering steps. The hierarchical clustering method used in the proposed method is bottom-up, which is continued to the threshold and creates clusters that represent coreference mentions. In clustering, the nodes are cut according to the weight of the edges so that the most probable coreference mentions can be recognized.

By using the graph, increase recall and find all of the coreference mention pairs. In each step, is given two mention graphs $G_i = \{m_{i\,1}; m_{i\,2}; \cdots; m_{i\,n}\}$ and $G_j = \{m_{j\,1}; m_{j\,2}; \cdots; m_{j\,n}\}$, that graph-pair encoder after combination of mention pair features, produce graph pair representations by applying a max-pooling operation to clusters. The architecture of the graph pair encoder is summarized in Figure 6.

The edges pruning step of the proposed method is determined based on the threshold. The initial training of the system detects this threshold. Finally, the last clustering determines which pairs are coreference and which are noncoreference.

The optimum threshold on the F1 value is seen in Figure 8. For this purpose, we use the B3 [35] coreference metric to obtain the optimal threshold level in the production of graph pair representations that consider the F1_muc value (Figure 8). For this purpose, only the B3 [35] criterion has been used. Although our system evaluation includes the MUC [36] and CEAF [33] metrics, we do not incorporate them into the graph pair representations because the MUC has the flaw of treating all errors equally, and the CEAF is slow to compute [14].

As seen in Figure 8, 0.8 is the desired value for the threshold.

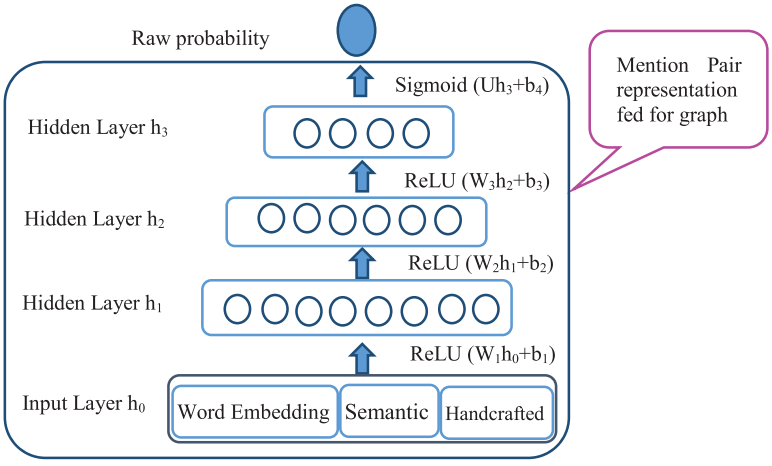

Figure 5 | Neural network architecture.

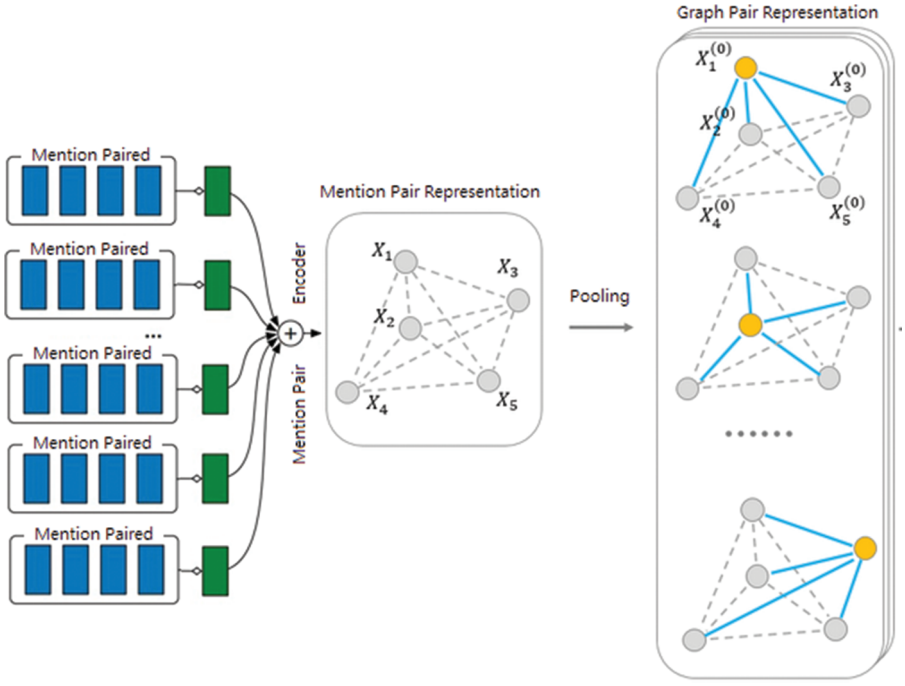

Figure 6 | The graph-based encoder.

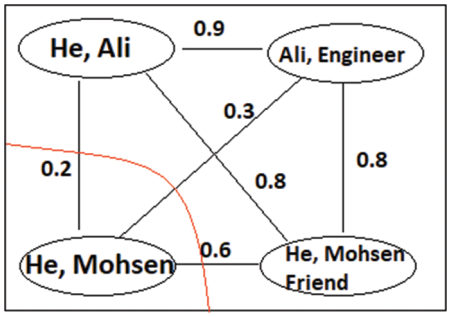

Figure 7 An example of graph pruning based on the weight of the edges.

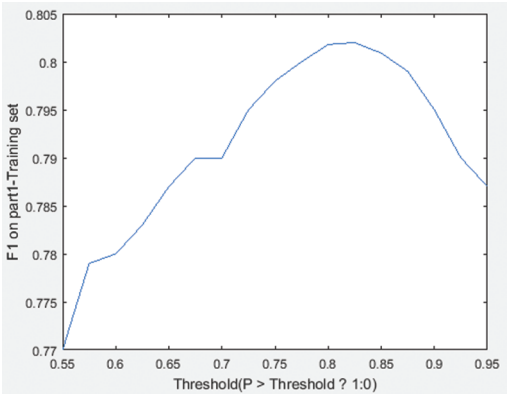

Figure 8 F1_muc value and different thresholds.

Figure 7 shows an example of graph coreference and their relationships. In this example, a coreference resolution algorithm decides to cut off the one node that noncoreference with others; as a result, it creates a chain coreference resolution with other nodes.

4. EXPERIMENTS AND RESULTS

We have runned experiments on the RCDAT corpus [8] and evaluate them with the MUC, B3, Ceafe, Ceafm and CoNLL metrics. We have trained the proposed method with hidden layers of sizes M1 = 800, M2 = M3 = 400.

4.1. Evaluation Metrics

The main problem in the evaluation of the coreference resolution systems is ambiguity about the number of existing mentions in the text. This problem becomes more significant than when the mentions resolved by the system do not match the mentions resolved by the golden standard. Therefore, the results obtained for different corpora will vary greatly. Also, most algorithms utilize postprocessing steps that will have a significant impact on the accuracy of the results. In this paper, we use the standard criteria that have been applied in various journals. These criteria are defined below.

4.1.1. MUC

First, the MUC conference defined MUC metric [36]. In MUC, the clusters obtained by the system and the standard golden are compared. The recall value is the ratio of the number of common links in the system and standard golden on the minimum standard golden links, and the precision value is the ratio of the number of common links in the system and standard golden on the minimum system links.

$$\mathrm{R} = \frac{\text{common links in system and standard golden}}{\text{minimum standard golden links}} \tag{1}$$

$$\mathrm{P} = \frac{\text{common links in system and standard golden}}{\text{minimum system links}} \tag{2}$$

4.1.2. B^3

To solve the weaknesses of the MUC, the B3 and CEAF have been introduced. In the B3, instead of links between clusters, the phrases themselves and the presence or absence of them in the classes are considered. The recall and precision values are the averages of R and P for each mention.

$$\mathrm{R} = \frac{\dfrac{\sum_{i=1}^{n}\left(\text{common mentions in system and standard golden}\right)}{\text{mentions in standard golden}}}{n} \tag{3}$$

$$\mathrm{P} = \frac{\dfrac{\sum_{i=1}^{n}\left(\text{common mentions in system and standard golden}\right)}{\text{mention in system}}}{n} \tag{4}$$

4.1.3. CEAF

In B3, system or standard golden mentions may have been considered more than once. To overcome this problem, CEAF [33] attempts to find the optimal one-to-one relationship between the system and the standard mentions, so CEAF is divided into two parts: entities-based and mention-based.

$$\frac{R}{P} = \frac{\text{common mentions in best one to one aligned system and standard golden}}{\text{mention in}\ \frac{\textit{standard golden}}{\textit{system}}} \tag{5}$$

4.1.4. CoNLL

The CoNLL metric is the average of the MUC, B3 and CEAF.

$$\text{CoNLL} = (\text{muc} + \text{b3} + \text{ceafm} + \text{ceafe})/4 \quad (6)$$

Despite the definition of more comprehensive evaluation metrics than the MUC, this metric is still used due to (1) comparison with the old systems that used only the MUC metric for evaluation and (2) the lack of a standard metric because none of the evaluation metrics has significant superiority over each other. Figure 9 represents the different types of evaluation metrics.

4.2. Results

In the following table, the results of the proposed method when compared with the state of art methods for coreference resolution are seen. To evaluate the proposed method, the Uppsala Test Corpus [9] has been used, which was labeled with the CoNLL2012 standard that contains 614 sentences and 16,274 words (26.5 words per sentence). The Uppsala Test Corpus 9 is quite comparable with the testing and development of the MUC-6 (the MUC-6 corpus has 13,000 tokens) and the MUC-7 (MUC-7 has 17,000 tokens). This corpus has been divided into four parts for which each part is a complete narrative. For training the model, the RCDAT corpus [8] was used. Table 3 shows a comparison of the proposed method with other related methods. The results show that the use of word embedding and semantic features improves the final value of F1 in the evaluation criteria, i.e., CEAF, MUC and B3.

As seen in Table 3, the proposed methods have been compared with essential methods in coreference resolution, and this comparison shows that the proposed method has higher accuracy than them.

For comparing the proposed method with other methods, as seen in Table 3, all of the methods were first compiled and trained on the corpus in [8], and then the Uppsala test corpus [9] was used to calculate the accuracy of each of them. In compiling the methods, some of the features cannot be used because they depend on language or require some of the corpora or preprocessing steps and cannot, therefore, be used for all languages.

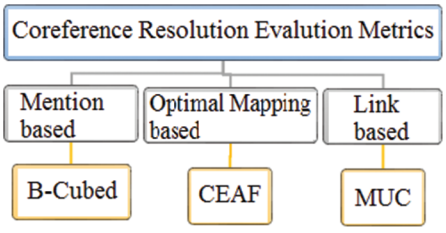

Figure 9 | Coreference resolution metric.

As we can see here (Table 3), the proposed method is more accurate than the other methods because the fully connected neural network, word embedding vector, and graph clustering method used in our proposed method improve its accuracy.

In [8], the authors used handcrafted features for coreference resolution. The extracted features in this method were string match, distance, m type, e type, acronym, first name mismatch and speaker detection. Then in this method, they used SVM classification to determine coreference resolution.

In [37], the authors extracted three categories of features from their corpus in order to calculate the cosine similarity of mention pairs; handcrafted features, word embedding features and entity linking features. With entity linking features, they first detect the mention and next find another mention related to Wikipedia page, then calculate their similarity with an LSTM that we also used to compile the redirect features, which was described in the semantic features section (3.2.3).

In [1], they initially recognized probable name entities with the LSTMs algorithm that we used along with the corpus in [8], which determined the name entities within it and then used the speaker and genre information from the metadata and the distance between the two spans and Glove embedding features to resolve the mention pair coreference.

Clark and Manning in [14,15] used 50 dimension word2vec, document genre, distance, speaker and string matching features in the feature extraction step. They next used a fully connected neural network to train the model. Then in [15], they used reinforcement learning to improve the neural network error. After training the model via the neural network, the authors of [14] used a clustering method to place coreference mentions into the same cluster.

In the proposed method, we used word embedding features and handcrafted features, and then trained the proposed method with a deep neural network and used hierarchical graph clustering to improve the proposed method's accuracy. As seen in Table 2, the comparison of the proposed method and other similar methods indicated that the proposed method improved the state art of coreference resolution algorithm accuracy from 59.56% to 64.54% in the Persian language.

Table 4 shows how each of the features used has improved the accuracy of the proposed system.

As seen in Table 4, the distance feature has a significant impact on the improvement of accuracy, and this is effective for the Persian and other languages.

Table 3 | Comparing the proposed method with another method.

	MUC (%)	B3 (%)	CEAFe (%)	CEAFm (%)	CoNLL (%)	Cluster-level Used	Word Embedding Used	FFNN
[8]	69.25	54.7	59.06	55.24	**59.56**	N	-	N
[37]	72.14	56.71	60.86	57.05	**61.69**	N	word2vec300d	Y
[1]	72.06	56.66	60.79	57.15	**61.66**	N	Glove300d	Y
[15]	71.79	56.12	60.27	56.77	**61.24**	Y	word2vec50d	Y
[14]	71.89	56.29	60.39	56.87	**61.36**	Y	word2vec50d	Y
Proposed method	**74.92**	**59.73**	**63.6**	**59.9**	**64.54**	**Y**	**Glove300d**	**Y**

Table 4 Comparison of the effect of each feature on the final accuracy of the proposed method.

Our Model	64.54%	0%
- Distance	61.34%	−3.2%
- Word embedding	62.14%	−2.4%
- Redirect featureand word embedding sentence	62.2%	−2.34%
- Mention type	60%	−2.2%
- Gender	60.4%	−1.8%
- String matching	60.7%	−1.5%
- Number agreement	60.8%	−1.4%

5. CONCLUSION AND FUTURE WORK

Providing a suitable coreference resolution method can solve the ambiguities in natural language processing systems such as news analysis and text summarization, etc.

In this paper, coreference resolution is first defined, then the problems and challenges of coreference resolution and actions used to solve these issues are explained. There are two rules-based and machine learning-based approaches for coreference resolution. The proposed machine learning-based algorithm in five steps are described as follows:

The first step: **Corpus Creation and Preprocessing**: In this step, the Persian coreference resolution for the RCDAT corpus [8] was described.

The second step: **Mention Pair Detection and Feature Extraction**: This is the process of mention pair detection expressed in [8], and in this paper, we described how to extract features from mention pairs. The extracted features were divided into three categories: semantic features, words embedding featuresand handcrafted features.

The third step: **Model Training**: In this step, we used a fully connected neural network to train the extracted features.

The fourth step: **Graph Formation and Coreference Resolution**: This step described how to form the graph and use an agglomerative hierarchical clustering method to cluster coreference mention pairs into a group.

Finally, the fifth step: **Evaluation**: In this step, the proposed method was compared with the other related methods in the Persian language, which indicated that the proposed method could increase the accuracy of the coreference resolution by 5%. However, it seems that the use of other semantic features such as the mentions sentence features or verb-related features in the mentions sentence can improve the accuracy of the proposed method.

CONFLICT OF INTEREST

Authors have no conflict of interest to declare.

ACKNOWLEDGMENTS

The authors thank the Iran Telecommunication Researches Center for sharing the Persian coreference resolution corpus as well as the Data Mining lab at Iran University of Science and Technology for the use of their resources and experiences.

REFERENCES

[1] K. Lee, L. He, M. Lewis, L. Zettlemoyer, End-to-end neural coreference resolution, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017, pp. 188–197.

[2] I. Chaturvedi, E. Cambria, R.E. Welsch, F. Herrera, Distinguishing between facts and opinions for sentiment analysis: survey and challenges, Inf. Fusion. 44 (2018), 65–77.

[3] J. Lu, V. Ng, Event coreference resolution: a survey of two decades of research, in IJCAI International Joint Conference on Artificial Intelligence, 2018, pp. 5479–5486.

[4] D. Tuggener, Coreference resolution evaluation for higher level applications, in Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, 2015, vol. 2, pp. 231–235.

[5] R. Mitkov, R. Evans, C. Orăsan, I. Dornescu, M. Rios, Coreference resolution: to what extent does it help NLP applications?, in International Conference on Text, Speech and Dialogue, 2012, pp. 16–27.

[6] J. Lu, V. Ng, Learning antecedent structures for event coreference resolution, in Proceedings - 16th IEEE International Conference on Machine Learning and Applications (ICMLA 2017), Cancun, Mexico, 2018, pp. 113–118.

[7] A. Haghighi, D. Klein, Simple coreference resolution with rich syntactic and semantic features, in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2009, vol. 3, pp. 1152–1161.

[8] Z. Rahimi, S. HosseinNejad, Corpus based coreference resolution for Farsi text, Signal and Data Processing, Iran, 17 (2020), 79–98.

[9] M. Seraji, J. Carina, M. Beata, N. Joakim, Uppsala Persian Dependency Treebank Annotation Guidelines, Technical Report, Uppsala University, 2013.

[10] Z. Liu, J. Araki, E. Hovy, T. Mitamura, Supervised within-document event coreference using information propagation, in Proceedings of the 9th Edition of the Language, Resources and Evaluation Conference (LREC 2014), Reykjavik, Iceland, 2014, pp. 4539–4544.

[11] H. Peng, Y. Song, D. Roth, Event detection and co-reference with minimal supervision, in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 2016, pp. 392–402.

[12] P.K. Choubey, R. Huang, Event coreference resolution by iteratively unfolding inter-dependencies among events, in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 2017, pp. 2124–2133.

[13] S. Pradhan, A. Moschitti, N. Xue, CoNLL-2012 shared task: modeling multilingual unrestricted coreference in OntoNotes, in Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task, Jeju Island, Korea, 2012, pp. 1–40.

[14] K. Clark, C.D. Manning, Improving coreference resolution by learning entity-level distributed representations, in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 2016.

[15] K. Clark, C.D. Manning, Deep reinforcement learning for mention-ranking coreference models, in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, 2016, pp. 2256–2262.
[16] H. Poon, P. Domingos, Joint unsupervised coreference resolution with Markov logic, in Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2010, pp. 650–659.
[17] F. Suchanek, G. Kasneci, G. Weikum, F. Suchanek, G. Kasneci, G. Weikum, Y.A. Core, F.M. Suchanek, G. Weikum, YAGO: a core of semantic knowledge unifying WordNet and Wikipedia, in 16th International World Wide Web Conference, WWW, Banff, Canada, 2007, pp. 697–706.
[18] S. Auer, C. Bizer, G. Kobilarov, J. Lehman, R. Cyganiak, Z. Ives, DBedpia: a nucleus for a web od open data, in: K. Aberer *et al.* (Eds.), The Semantic Web, Lecture Notes in Computer Science, vol. 4825, Springer, Berlin, Heidelberg, Germany, 2007, p. 722.
[19] G.A. Miller, WordNet: a lexical database for English, Commun. ACM. 38 (1995), 39–41.
[20] M. Asgari, A. Hadian, B. Minaei-Bidgoli, FarsBase: the persian knowledge graph, Semant. Web J. (2019).
[21] O. Uryupina, M. Poesio, C. Giuliano, K. Tymoshenko, Disambiguation and filtering methods in using web knowledge for coreference resolution, in: C. Boonthum-Denecke, P.M. McCarthy, T. Lamkin (Eds.), Cross-Disciplinary Advances in Applied Natural Language Processing, IGI Global, Hershey, PA, USA, 2012, pp. 185–201.
[22] T. Mikolov, W.-T. Yih, G. Zweig, O'Sullivan- classification of lumbopelvic pain disorders, 2013, pp. 9–14.
[23] P. Jeffrey, S. Richard, D.M. Christopher, GloVe: global vectors for word representation, in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, vol. 19, pp. 417–425.
[24] W.M. Soon, H.T. Ng, D.C.Y. Lim, A machine learning approach to coreference resolution of noun phrases, Comput. Linguist. 27 (2012), 521–544.
[25] H. Ji, R. Grishman, Knowledge base population: successful approaches and challenges, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 2007, vol. 1, pp. 1148–1158.
[26] C. Duncan, L.-W. Chan, H. Peng, H. Wu, S. Upadhyay, N. Gupta, C.-T. Tsai, M. Sammons, D. Roth, UI CCG TAC-KBP2017 submissions: entity discovery and linking, and event nugget detection and co-reference, in Proceedings of Text Analysis Conference (TAC 2017), 2017.
[27] C. Nicolae, G. Nicolae, Bestcut: a graph algorithm for coreference resolution, in Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 2006, pp. 275–283.
[28] P. Denis, J. Baldridge, Specialized models and ranking for coreference resolution, in Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 2008, pp. 660–669.
[29] A. McCallum, Toward Conditional Models of Identity Uncertainty with Application to Proper Noun Coreference, Books.Nips.Cc., 2005, pp. 208–222.
[30] B. Yang, C. Cardie, P. Frazier, A hierarchical distance-dependent bayesian model for event coreference resolution, Trans. Assoc. Comput. Linguist. 3 (2015), 517–528.
[31] S. Jiang, Y. Li, T. Qin, Q. Meng, B. Dong, SRCB Entity Discovery and Linking (EDL) and event nugget systems for TAC 2017, in Proceedings of Text Analysis Conference (TAC 2017), 2017.
[32] A. Rahman, V. Ng, Coreference resolution with world knowledge, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, OR, USA, 2011, vol. 1, pp. 814–824.
[33] X. Luo, On coreference resolution performance metrics, in Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2005, pp. 25–32.
[34] H. Poostchi, E. Zare Borzeshi, M. Piccardi, BiLSTM-CRF for persian named-entity recognition; ArmanPersoNERCorpus: the first entity-annotated persian dataset, in The 11th Edition of the Language Resources and Evaluation Conference (LREC), Miyazaki, Japan, 2018.
[35] A. Bagga, B. Baldwin, Algorithms for scoring coreference chains shortcomings of the MUC-6 algorithm, in Proceedings of the 1st International Conference on Language Resources and Evaluation, Granada, Spain, 1998, pp. 563–566.
[36] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, L. Hirschman, A model-theoretic coreference scoring scheme a problematic case, in Proceedings of the Sixth Message understanding Conference (MUC-6), Morgan Kaufmann Pu, San Francisco, CA, USA, 1995, pp. 45–52.
[37] A. Sil, G. Kundu, R. Florian, W. Hamza, Neural cross-lingual entity linking, in Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 5464–5472.