# Distributed Representation of Entity Mentions Within and Across Multiple Text Documents

Aliakbar Keshtkaran*, Siti Sophiayati Yuhaniz, Mohammad Reza Rostami

*Advanced Informatics Department,*
*Razak Faculty of Technology and Informatics*
*Universiti Teknologi Malaysia*
*Jalan Sultan Yahya Petra, 54100, Kuala Lumpur*

*aakeshtkaran@live.utm.my*
*sophia@utm.my*
*rostami2007@gmail.com*

**Article history**

Received:
4 Nov 2019

Received in revised form:
18 Nov 2019

Accepted:
4 Dec 2019

Published online:
25 Dec 2019

*Corresponding author:
aakeshtkaran@live.utm.my

***Abstract***

*Regarding to the importance of entities as a base of information for several NLP applications, Cross-Document Coreference Entity Resolution (CDCR) provides techniques for the identification of textual mentions of entities and clustering co-referent mentions across multiple documents. In such context, while prior works employ Knowledge Bases (KB) as a structured information resource to enrich the context of mentions, however these methods have limitations with KB's unknown entities, with effects on the accuracy and performance of the task. Accordingly, this paper presents a new approach to improve the state-of-the-art by concentration on the knowledge provided by the input text of the mentions, regardless of any external knowledge resource. For this purpose, we first construct the context of mentions using the sequence of informative words around the mention (known as content-words). Furthermore, by abstraction of the mention vector representation to a limited size using an artificial neural network technique of continuous representation of words (i.e. Word2Vec), we reduce the computational cost of the co-referring mentions sub-task. By analyzing the results of experiments with two datasets, significant gains in the accuracy of CDCR as well as run-time efficiency are achieved, compared to the best prior methods.*

***Keywords:*** *Coreference Resolution, Cross-Document Coreference Resolution, Distributed Representation of Words, Information Extraction, Natural Language Processing*

## 1. Introduction

The mainstream part of the information produced by digital devices is globally expressed in the form of natural language text such as web pages, news articles, medical records, government documents, social media, etc. Such form of data is totally termed unstructured versus structured data that is normalized and stored in a database somehow that each record is divided from other records and relevant features are associated to it. Information Extraction (IE) systems concern about automatically extraction of information from unstructured/semi-structured data [1]. For this purpose, to extract the locked information in unstructured text, Natural Language Processing (NLP) is used to discover and produce structured information.

Among various sub-tasks of NLP Coreference Resolution (CR) is essential to identify entity mentions in the text and resolve them into equivalent classes [2-6].

* *Corresponding author: aakeshtkaran@live.utm.my*

In such context, an entity can be a real-world person, organization, or place, which is referred to, by a mention, i.e. a word or phrase referring to such an entity (Fig. 1).

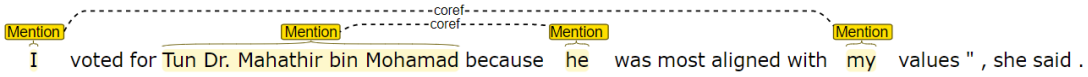


**Fig. 1. An Example of Coreference Resolution**

Expanding the scope of CR to process a collection of documents and resolving the entities across the documents leads to Cross-Document Coreference Resolution (CDCR) [7-11]. CDCR plays a key role for several high-end NLP applications such as Automatic Knowledge Base Construction, Question Answering System, Automatic Text Summarization and Search Engines [12],[8].

CDCR involves various subtasks, from identifying entities and mentions to co-referring the mentions. The overall objective is to group mentions such that mentions referring to the same entity are in the same group and no other entities are included. Mentions referring to the same entity are termed "coreferent" [11]. Accordingly, the challenges of CDCR can be mentioned as below:

**Efficient Context Detection:** Capturing the similarities of mentions by efficient state-of-the-art methods for CR in a single document is conducted by syntactic and linguistic features [13], multi-phase sieves [3] or entity-level distributed representation [14-15]. Such methods provide high accuracy for ICR, however, CDCR approaches using pairwise mention comparisons are computationally expensive make them for web scale CDCR tasks. Moreover, deep understating of mention contexts within and across documents are required for two main CDCR challenges, i.e. Entity Disambiguation and Name Variation. While Entity Disambiguation task is to distinguish between different entities with similar text (e.g., several entities with name "John"), Name Variation task is to co-refer different surface forms of an entity name (e.g., name "Jonathan" abbreviated to "Jon"). To enrich the knowledge using the relational information of entities, mention attributes extracted from Knowledge Bases (KB's) – i.e. knowledge databases used for knowledge sharing, are employed in recent works [16],[8],[2]. However, such featurization approaches cannot be reliable because the construction of KB's depends on CDCR results. Accordingly, such a recursive dependency of CDCR and KB's suffers from efficiency issues specifically for detecting the context of long-tail entities (i.e. entities which have a less complete profile in KB's) or unknown entities in KB's.

**Scalable Entity Embeddings:** Since machine learning algorithms cannot work with raw text directly; the text must be converted into numbers, specifically, vectors of numbers. Word embedding is the collective name for a set of language modeling and feature learning techniques in NLP, where words or phrases from the vocabulary are mapped to vectors of real numbers. A commonly used model of word embedding for NLP tasks is Bag-of-Words (BoW) which the frequency of occurrence of each word is used as a feature of word vector [17]. Considering the size of the feature vectors by BoW which depends on the vocabulary size of the document collection, by increasing in the number of mentions, the word embedding approaches based on BoW meet the scalability issues.

**Efficient Entity Clustering:** CDCR can be viewed as a clustering problem of entity mentions based on their context similarities. Linguistics features of mentions

are commonly employed to compute the similarity between their contexts These features can be syntactic path between mentions, their semantic compatibility, and the distance between them [18],[13]. Word embeddings provide the mapped numerical values of words in the vocabulary. For the task of clustering, CDCR also meets two challenges. (1) Often the number of underlying entities and their identities are not known. (2) Unlike inference in other language processing tasks that scales linearly in the size of the corpus, the hypothesis dimension of features for coreference across documents grows super exponentially with the number of mentions. . However, interdependencies of entities and their context are ignored in standard clustering which leads it to suffer from high computational complexity.

This research focuses on a new approach to cast entities of documents in an optimized fixed-length dimension size of vector with considering the sequence of words and phrases in the document to outperform the accuracy of the task. This approch can also improve the performance of the clustering stage of CDCR due to the small size of the dimension of vectors compared with other approaches. Accordingly, in Section 2 related words shall be discussed. The approach of considered model is discussed in Section 3. Evaluation experiments on different benchmark datasets and results using different evaluation measurements are discussed on Section 4. Finally, the conclusion of the study is presented in Section 5.

## 2. Related Work

To handle the problems mentioned in Introduction, several solutions are proposed by researchers. For CDCR task, the first study presented by Bagga and Baldwin [19] which they used the Vector Space Model (VSM) to disambiguate entities across documents. Later, Gooi and Allan [20] presented three other models for CDCR based on the incremental vector space, KL divergence (the probabilistic approach), and a hierarchical clustering approach. More complicated models were presented by researchers later, established on one of the three main modelling approaches: graph-based model [9],[21] probabilistic model [11],[22], and clustering-based model [12],[8],[23],[24],[10]. Nonetheless, they have not fully paved the way to satisfying results of resolving entities across documents regardless of the size of document collection.

Although a few of works have mainly concentrated on scalability of data size [8],[10],[25], however they suffer from problems like very large dimension size of feature space or inefficient context detection. The former is caused by the applied techniques for word vectorization like Bag-of-Words [8] which the dimension of word embeddings grows with increasing the vocabulary size of document collection. For the latter, while Dutta and Weikum [8] considered an efficient context detection approach called Knowledge Enrichment, however it has the problem of relying on Knowledge Bases (KB's). Note that since the KB's are constructed based on CDCR results, this recursive dependency of CDCR and KB's can suffer the data processing task specifically with long-tail entities or unknown entities in KB's.

Accordingly, while a few researches have been conducted in the area of CDCR, there are still open issues related to the efficiency and scalability of the CDCR task. In order to address this goal, the scalability in size of the data and its dimension as well as efficient and precise context detection without conducting any contextual

enrichment based on KB's should be considered. Therefore, the current research aims to investigate a more accurate solution compared to previous works which can outperform the accuracy and performance of the CDCR.

## 3. Approach

In the development of CDCR the final goal is to get the best possible grouping of entity mentions in which two entity mention belong to the same group if and only if they refer to the same underlying entity. This objective naturally leads to the design of some Machine Learning (ML) techniques or combine several ML techniques for the problem to be solved. The achievement of such a system should be improvement in the accuracy of the CDCR task, as well as the performance.

In our proposed model we assume that an input set of documents $D = \{d_1, d_2, \cdots\}$ with a set of the entity mentions of all documents $M = \{m_{11}, m_{12}, \cdots, m_{11}, m_{11}, \cdots\}$, where $m_{ij} \epsilon d_i$. As output, the model computes and equivalence relation over M with equivalence classes $C_l$ where $C_l \cap C_n = \emptyset$ (for $k \neq n$) and $\cup_k C_l = M$. The number of desired classes is needed to be computed by the algorithm, since it is unknown in advance. The model majorly consists of three operational stages (Fig. 2):

**Pre-Processing:** Given an input corpus of text documents, initially text cleaning is run to cast them into plain text. It then detects the entity mentions in the text and their lexical type such as person, organization, or location. The input of Intra-Document Coreference Resolution (ICR) is formed which is a collection of text documents with tagged identified mentions. In this step, Local chains of coreference mentions are resolved by the state-of-the-art CR tool. The annotated texts and local chains of co-referent mentions form the in input of the second stage.

**Entity Vectorization:** In this stage, for each of the local mention groups $(\{m_{ij}\})$ obtained in the previous stage, a scalar vector as its representative is constructed to be clustered in the next stage. Initially, the distributed representation of each mention (i.e., word or phrase) is generated based on its informative context words known as Content Words. In the next step, the combination of mention vectors of each chain, forms the vector of the chain. The output of this stage is a collection of scalar vectors representing entities respectively.

**Clustering:** Through the proper distribution of mention chain vectors in the vector space, obtained in the previous stage, a clustering algorithm is used in the model to group the mentions based on their similarities. Specifically, a density-based clustering algorithm called DBSCAN (Density-Based Spatial Clustering for Applications with Noise) is used in the model. The output of this stage is the cross-document coreference equivalence classes of entity mentions.


**Fig. 2. The Stage of Cross-Document Coreference Resolution Model**

### 3.1 Pre-processing

The initial step of our model is pre-processing the input documents. Given an input corpus of text documents, it initially runs text cleaning to remove hypertexts

or non-alphanumerical characters using tools like Boilerpipe [36]. The Stanford CoreNLP tool [37] is used to analyze each document separately to detect and tag the mentions of the document's text. The Stanford NER Tagger [26] is employed in the next step to tag mentions with their lexical types like person, name organization. This step is the intra-document pre-processing and forms the input of Intra-Document Coreference Resolution (ICR), a collection of text documents, D with identified and tagged mentions M.

In the next step, Local chains of coreference mentions are resolved by the Stanford CoreNLP tool based on multi-sieve algorithm from Stanford [13],[3],[27],[28]. The text, tagged mentions, and the extracted coreference chains of each document are then passed to the second stage. While it is possible that some errors like irrelevant mention chains are produced by the ICR step, however, outperforming the results of ICR is out of scope of this research.

### 3.2 Entity Vectorization

To form the sequence of extracted entities in previous stage surrounding with their context four steps and their details are defined in six part. The sequential model of these steps is presented in Fig. 3.

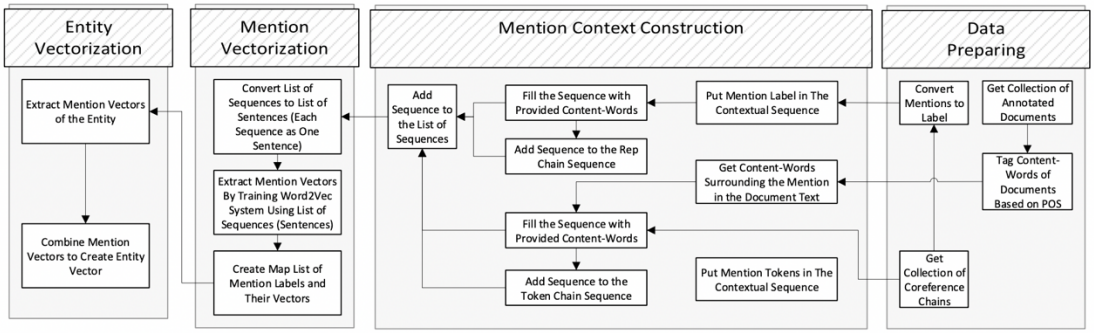


**Fig. 3. Four Steps of Entity Vectorization with Details**

**Mention Representation:** For two reasons, mention as a word or phrase, cannot be used directly in the vectorization process. (1) A phrase mention consists of more than one word and so, during the tokenization step of the input they will be considered as distinct words. Since the goal of this vectorization approach is to compute the distribution of the mention–i.e. with the set of its words and not as distinct words, among the words around the mention known as the context of the mention. Based on this, to convert the mention to a single token, it is necessary to replace the mention with a single word representative. (2) Although some of the mentions have string matching, they should be considered as different tokens for the vectorization process. This is because of different contexts which the mentions are composed by, and so they should be trained separately.

Based on these two reasons, the mentions are replaced with a single word label as the representative. Labels are generated by a simple function which combines these parameters to ensure the uniqueness of the label. (1) Mention String (e.g. "John Smith"). (2) Mention Location in the text (e.g. 3,2: Sentence 2, Token 3). (3) The mention's container Document ID (e.g. Document 3). By using these

parameters, the representative label of the mention is generated which will be used for the training step of the vectorization.

**Content Words Tagging:** One of the exports of the pre-processing stage is part-of-speech (POS) tagging of document words. POS tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context—i.e., its relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc. Using the POS tag of the word, the word in the text among other words, can be categorized as Lexical Word or Grammatical Word. In English grammar and semantics, a content word is a word that conveys information in a text or speech act, also known as a lexical word, lexical morpheme, substantive category, or contentive. Contrast with function word or grammatical word.

"All morphemes can be divided into the categories lexical [content] and grammatical [function]. A lexical morpheme has a meaning that can be understood fully in and of itself. Nouns, verbs, adjectives, and adverbs are typical kinds of lexical morphemes. Grammatical morphemes, on the other hand can be understood completely only when they occur with other words in a sentence." [29].

Relying on the above definition and the role of the content words to deliver the required information of the text, all the content words of the document are tagged based on their part of speech tag. For this purpose, all nouns, verbs, adjectives and adverbs are tagged as content word. Additionally, despite pronouns are considered as grammatical word, but they should be considered as content word. The latter is because of the role of pronouns in the task of coreference resolution which some mentions may be pronoun. These content words are going to be used in the next steps for constructing the word sequence of the context of each mention.

Context Window Size: The goal of mention vectorization is to compute the distribution of the mention among its context. The Context of the mention is created by the content words around the mention. The number the words around the mention can be considered from two aspects:

(1) The number of the words around the mention as the input of vectorization neural network. The neural network of vectorization is trained based on the possibility of the occurrence of the mention between the input words. This aspect of window size can be called Training Window Size (TWS).

(2) The number of the words around the mention which should be trained by the vectorization neural network model. This aspect defines how many words around the mention are needed to be trained to compute the distribution of possibility of their occurrence among their surrounding words which this also is characterized by the training window size referring to aspect one. This aspect of window size can be called Context Window Size (CWS). The number of training window size can be set based on the experiment results. For the Context Window Size, it can be set by many approaches.

Two suggested approaches by this model is first, based on a multiplication of the Training Window Size. The second approach is setting the CWS based on a fixed number of sentences around the mention, including the containing sentence of the mention. The latter can be considered more acceptable, since it is expected that the mention context is defined by its surrounding sentences. By considering the three

mentioned settings, the mention and its context is ready for the construction of the words sequences to be trained later in the vectorization network.

Contextual Word Sequence Based on Document Text: Using the prepared representative label of the mention and its context based on the content words, the input is ready to construct the sequence of the words for training using the vectorization neural network. For this purpose, two approaches are available for the construction task.

(1) Based on Mention's Representative; To put the mention's representative (described in part one of this stage) in the center of the sequence and fill around it with its surrounding content words respectively, limited to the number of CWS (as described in part three of this stage).

(2) Based on Mention's Tokens; Same as number 1, instead of putting the mention's representative, putting the mention in its original form with its tokens in the center and filling the sequence same as number 1 respectively. Although its mentioned in part two, which the mention is replaced with its single word label as the representative, however a sequence by the mention's tokens can be generated. The latter can be described as follow: The construction of a sequence with the mention tokens is required due to the importance of the lexical similarity of mentions which mentions with the same string should be trained to satisfy the following issue: During the construction of the sequences, when the sequence is made by the mention and its context (primary mention), it is possible a secondary mention be in the context of the primary mention of the sequence (each sequence belongs to a mention and filled with the primary mention and its context) and when it's added to the sequence it is in its original form with its tokens (contrary of mention's representative). This state leads to a situation which the secondary mention tokens are trained during the process. The sequences generated in this step are added to the list of sequences. This list of sequences is completed in the next step to be prepared for training using vectorization neural network.

Contextual Word Sequence Based on Mention Chain: The extracted local mention chains in the pre-processing stage are used in two ways. The first is to get the desired mentions for vectorization. The second is to generate sequences of the mention chain by its mentions and their context. For this purpose, first, the mentions in the chain are sorted based on their occurrences in the text. Following this, a sequence for the mention chain is created following the same approach for sequence construction based on the document text. This aim is achieved by creating the sequence of each mention of the chain and joining the sequences of the mentions followed by their order in the chain together the created to sequence of the mention chain. It also should be noticed that, same as the previous step, two set of the sequences based on mention's representative and mention tokens are generated. Finally, the generated sequences are added to the list of sequences. Until this step, the sequences of the mentions are created, and the full list of the sequences is prepared. In the next step, these sequences are going to be trained using the vectorization neural network to generate the distributed representation vector of the mentions.

Vectorization of the Entities: Word embedding is one of the most popular representation of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words. Word2Vec is one of the most popular technique to learn word embeddings using

shallow neural network. Word2vec is a two-layer neural network that processes text. Its input is a text corpus and its output are a set of vectors: feature vectors for words in that corpus. Word2vec was created by a team of researchers at Google [30]. Embedding vectors created using the Word2vec algorithm have many advantages compared to earlier algorithms such as Bag-of-Words (BoW). In this step, sequences of the mentions, created in the previous step are fed into Word2Vec system to generate the vectors of the mentions. Using Word2Vec, this task is done through two sub-steps. First, the vectors of mention tokens are created, and then the vectors are combined using a combining function. Second, the vectors of the representative labels of mentions, surrounding by its context words in the sequence are created. By combining these vectors of the mention using a combining function, the vector of the mention is created. Since, each mention chain obtained from document is representing an entity, the vector of the entity is created by combining the vector of its mentions in the chain. Having entity vectors with fixed and limited dimension size makes the model proper to use a clustering algorithm to group similar entities extracted from documents based on their similarities.

**Algorithm 1:** Entity Vectorization

Require: Text T, Set E of entities (Mention Groups from Pre-processing), Combining Function F, Configuration C of Word2Vec, Types T2 of Content Words, Size S1 of Context Window Size, Size S2 of Training Window Size
Ensure: Fixed-length Vectors of E (Mention Groups)
1: for each entity, e ∈ E do
Content Words (CW): extract content words of type T2 from T
2: for each mention, m ∈ M of e do
3: Mention Representation (MR): convert m to single unique word as its representative
4: Sequences Based on Document Text: Sequence-1 ← MR & Sequence-2 ← m
5: Right Context: Sequence-1 & Sequence-2 ← put S2 number of CW after m in T right of m
6: Left Context: Sequence-1 & Sequence-2 ← put S2 number of CW before m in T left of m
7: end for
8: Sequences Based on Mention Chain: Sequence-3 ← All Sequence-1 of sorted M & Sequence-4 ← All Sequence-3 of sorted M
9: end for
10: Input of Word2Vec: List all Sequences of M of All E
11: Mention Vector: Produce Vector of Mentions Representatives Using Word2Vec Based on C
12: for each entity, e ∈ E do
13: Entity Vector: Combine M Vectors Using F
14: end for
15: Output List of Vectors of all E

### 3.3 Clustering

Through the proper distribution of entity vectors in the vector space, obtained in the previous stage, a machine learning-based technique can group the entities based on their similarities. For this purpose, since the possible entities of result are apriori unknown, so the system cannot be trained by previously known groups which it leads the selection of machine-learning technique to clustering. The clustering algorithm also is needed to be able to detect any number of clusters, due to the unknown number of entities. In our experiments, DBSCAN algorithm (Density-Based Spatial Clustering for Applications with Noise) is used to cluster entities

based on their distribution in the vector space. DBSCAN is a data clustering algorithm proposed by in 1996 [31]. It is a density-based clustering non-parametric algorithm: given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. It is proposed in this experiment that there is no outlier and any single entity without any neighbor in the vector space is considered as a cluster. The output of this stage is the predicted clusters of entities according to the input corpus.

## 4. Evaluation

### 4.1 Benchmark Datasets

To evaluate and compare the accuracy of our model against state-of-the-art baselines, we run the experiments with the following two available benchmarking datasets.

**John Smith Corpus:** This dataset [19] is a highly ambiguous dataset which consisted of 197 articles from 1996 and 1997 editions of the New York Times. The relatively of common name `John Smith' used to find documents that were about different individuals in the news. The corpus is a combination of multiple mentions of “John Smith” related to 35 different real person entities. All John Smiths in a document refer to the same person. Since most of the entities in this dataset are longtail entities which make them unknown to any KB, it is a suitable small-scale benchmark dataset to evaluate any CDCR model.

**WePS-2 Collection:** This collection is used in the Web People Search 2 competition [32]. Using Yahoo search for 30 different people, the top 150 Web search results create the collection of 4,500 documents of this dataset. Each person name of the dataset is annotated with the ground truth files by human annotators. This collection offers a real corpus, which can test a system to resolve varying ambiguous personal names in different domains.

In our experiments, only lexical type of person is considered which is the most demanding type of entities in the datasets. We conducted all of the experiments on a 2 core Intel i7 2.40 GHz processor with 12 GB RAM running Windows 10.

### 4.2 Evaluation Measures

Among different evaluation measurements for CDCR task, two established measures are employed in our experiments to assess the accuracy of our CDCR model. These two measures are selected because of using them in multiple CDCR research works and their meaningful interpretable results.

**B3 F1 score [33]:** In this measure while precision is the ratio of the number of correctly reported coreferences (for each mention) to the total number, recall is the fraction of actual coreferences correctly detected. Both the final precision and recall are computed by averaging over all mention groups. The F1 score is finally computed as a harmonic mean of precision and recall of the final equivalence classes.

**φ3-CEAF score [34]:** This measure computes precision, recall, and F1 scores using a 1-to-1 mapping way. The approach is to measure the best 1-to-1 mapping between the equivalence classes obtained and those in the ground truth. In this measure the highest mention overlap is displayed by the best mapping of ground-truth to output classes.

### 4.2 Results

#### 4.2.1 John Smith Corpus

Best published results for this dataset are compared with our model results in Table 1. For training dataset, we randomly selected 30% documents and the rest are used as the test set. Our model achieves B3 based F1 accuracy of 85.93% on this dataset as the best result and an average of 81.32% for 10-fold runs. While Stream [10] and Inference [11] reach only 69.7% and 66.4% resp, the best previous result by Dutta and Weikum (2015b)[8] obtains 75.21% using the model most similar to ours, while their best model (which knowledge enrichment is done by a Knowledge Base) achieves 76.47%. Our model also achieves φ3-CEAF score of 79.57% which outperforms results over prior methods. The runtime of clustering of our model was only around 1.5 seconds comparing result of [8] shows a faster process.

#### 4.2.2 WePS-2 Collection

Our model is also compared against the best methods reported in [8],[35] on the WePS-2 collection. Our model achieves a B3 based F1 score of 78.28% and a φ3-CEAF score of 72.97% (Table 2). Compared to previous works our results are less than best reported results of [8] and [35] which are reached 83.48% and 83.8% resp. It should be noticed that the best previously reported results are based on using KB's and since WePS-2 dataset is collected from web, it is obvious that KB can improve the result. It is reasonable when we compare our result with those previously reported without using KB's which is 63.5% of B3-F1 for [8] and 75.7% of B3-F1 for [35]. The runtime of the model on WePS-2 corpus was about 23 seconds.

**Table 1. Results on Datasets**

| | John Smith | | WePS-2 | |
|---|---|---|---|---|
| | B3-F1 | φ3-CEAF | B3-F1 | φ3-CEAF |
| Our Model (Regardless KB) | 85.93% | 79.57% | 78.28% | 72.97% |
| (Emami, 2019) / Regardless KB | - | - | 75.7% | - |
| (Dutta & Weikum, 2015b) / Regardless KB | - | - | 76.9% | - |
| (Singh et al., 2011) | 69.7% | - | - | - |
| (Rao et al., 2010) | 66.4% | - | - | - |
| (Dutta & Weikum, 2015b) | 75.21% | 69.89% | 83.48% | (74.02%) |
| (Emami, 2019) | - | - | (83.8%) | - |

## 5. Conclusion

In this research we have presented our model for cross-document coreference resolution (CDCR) task. It performs density-based clustering over fixed-length dimension size of entity vectors obtained from a distributed representation of entities of documents. The effective sequence of words extracted from mentions and their context achieved by including content words of the context. The casting approach for single unique word as mention representative is the key point of this model to process mentions with any number of tokens effectively. It is encouraging to note that our approach, using only the input documents regardless of any external knowledge, performs competitively with related works with improve accuracy and performance of the task. However, due to the small size of the datasets, we require further experiments for future works.

## 6. References


[1] McCallum, A. (2005). Information extraction: Distilling structured data from unstructured text. Queue, 3(9), 48-57.
[2] Hajishirzi, H., Zilles, L., Weld, D. S., & Zettlemoyer, L. S. (2013). Joint Coreference Resolution and Named-Entity Linking with Multi-Pass Sieves. Paper presented at the EMNLP.
[3] Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., & Jurafsky, D. (2011). Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. Paper presented at the Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task.
[4] Màrquez, L., Recasens, M., & Sapena, E. (2013). Coreference resolution: An empirical study based on SemEval-2010 shared task 1. Language resources and evaluation, 47(3), 661-694.
[5] Ng, V. (2016). Advanced Machine Learning Models for Coreference Resolution. In Anaphora Resolution (pp. 283-313): Springer.
[6] Rahman, A., & Ng, V. (2011). Ensemble-based coreference resolution. Paper presented at the IJCAI Proceedings-International Joint Conference on Artificial Intelligence.
[7] Beheshti, S. M. R., Benatallah, B., Venugopal, S., Ryu, S. H., Motahari-Nezhad, H. R., & Wang, W. (2017). A systematic review and comparative analysis of cross-document coreference resolution methods and tools. Computing, 99(4), 313-349. doi:10.1007/s00607-016-0490-0
[8] Dutta, S., & Weikum, G. (2015b). Cross-document co-reference resolution using sample-based clustering with knowledge enrichment. Transactions of the Association for Computational Linguistics, 3, 15-28.
[9] Ngomo, A.-C. N., Röder, M., & Usbeck, R. (2014). Cross-document coreference resolution using latent features. Paper presented at the Proceedings of the Second International Conference on Linked Data for Information Extraction-Volume 1267.
[10] Rao, D., McNamee, P., & Dredze, M. (2010). Streaming cross document entity coreference resolution. Paper presented at the Proceedings of the 23rd International Conference on Computational Linguistics: Posters.
[11] Singh, S., Subramanya, A., Pereira, F., & McCallum, A. (2011). Large-scale cross-document coreference using distributed inference and hierarchical models. Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1.
[12] Baron, A., & Freedman, M. (2008). Who is who and what is what: experiments in cross-document co-reference. Paper presented at the Proceedings of the Conference on Empirical Methods in Natural Language Processing.
[13] Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., & Jurafsky, D. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational linguistics, 39(4), 885-916.
[14] Clark, K., & Manning, C. D. (2016a). Deep reinforcement learning for mention-ranking coreference models. arXiv preprint arXiv:1609.08667.
[15] Clark, K., & Manning, C. D. (2016b). Improving coreference resolution by learning entity-level distributed representations. arXiv preprint arXiv:1606.01323.
[16] Dutta, S., & Weikum, G. (2015a). C3EL: A joint model for cross-document co-reference resolution and entity linking. Paper presented at the Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.
[17] Goldberg, Y. (2017). Neural network methods for natural language processing. Synthesis Lectures on Human Language Technologies, 10(1), 1-309.
[18] Haghighi, A., & Klein, D. (2009). Simple coreference resolution with rich syntactic and semantic features. Paper presented at the Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3.
[19] Bagga, A., & Baldwin, B. (1998b). Entity-based cross-document coreferencing using the vector space model. Paper presented at the Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1.
[20] Gooi, C. H., & Allan, J. (2004). Cross-document coreference on a large scale corpus. Paper presented at the Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004.
[21] Rahimian, F., Girdzijauskas, S., & Haridi, S. (2014). Parallel community detection for cross-document coreference. Paper presented at the Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on.
[22] Singh, S., Wick, M., & McCallum, A. (2010). Distantly labeling data for large scale cross-document coreference. arXiv preprint arXiv:1005.4298.
[23] Finin, T., Syed, Z., Mayfield, J., McNamee, P., & Piatko, C. D. (2009). Using Wikitology for Cross-Document Entity

Coreference Resolution. Paper presented at the AAAI Spring Symposium: Learning by Reading and Learning to Read.
[24] Mayfield, J., Alexander, D., Dorr, B. J., Eisner, J., Elsayed, T., Finin, T., . . . McNamee, P. (2009). Cross-Document Coreference Resolution: A Key Technology for Learning by Reading. Paper presented at the AAAI Spring Symposium: Learning by Reading and Learning to Read.
[25] Wick, M., Singh, S., & McCallum, A. (2012). A discriminative hierarchical model for fast coreference at large scale. Paper presented at the Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1.
[26] Finkel, J. R., Grenager, T., & Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. Paper presented at the Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics.
[27] Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., & Manning, C. (2010). A multi-pass sieve for coreference resolution. Paper presented at the Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing.
[28] Recasens, M., de Marneffe, M.-C., & Potts, C. (2013). The life and death of discourse entities: Identifying singleton mentions. Paper presented at the Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.
[29] Murray, T. E. (1995). The structure of English: Phonetics, phonology, morphology: Allyn and Bacon.
[30] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[31] Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. Paper presented at the Kdd.
[32] Artiles, J., Gonzalo, J., & Sekine, S. (2009). Weps 2 evaluation campaign: overview of the web people search clustering task. Paper presented at the 2nd web people search evaluation workshop (WePS 2009), 18th www conference.
[33] Bagga, A., & Baldwin, B. (1998a). Algorithms for scoring coreference chains. Paper presented at the The first international conference on language resources and evaluation workshop on linguistics coreference.
[34] Luo, X. (2005). On coreference resolution performance metrics. Paper presented at the Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing.
[35] Emami, H. (2019). A Graph-based Approach to Person Name Disambiguation in Web. ACM Transactions on Management Information Systems (TMIS), 10(2), 4.
[36] https://boilerpipe-web.appspot.com/
[37] Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60