DocOIE: A Document-level Context-Aware Dataset for OpenIE

Amira Chebbi
amira98chebbi@gmail.com

Abstract

Extracting information from textual sources plays a critical role in many areas, especially in NLP tasks.The open information extraction itself is the fact of extracting relation tuples without referring to a predefined pattern.An information extraction based only on the sentence level neglect relevant information that may exist in the surrounding context.An approach that refers to the contextual information is discussed in this paper. [DocOIE, a new Dataset composed of 800 manually annotated sentences with context awareness, is presented in this work]DocOIEAbstract.[This DocOIE is used in the evaluation process of the DocIE model.In fact, the presented system enables the extraction of relations with a document-level context awareness]DocIEAbstract.

1 Introduction

Open Information extraction is the task of extracting structured tuples from unstructured data. In this process, a user intervention is not required, and the extraction is fully domain independent.Furthermore, the openIE systems like text runner (Yates et al., 2007) support fast, interactive and powerful relational queries. For this reason, openIE was used in a wide range of NLP tasks like question answering (Khot et al., 2017) (Gashteovski et al., 2020) and text summarization (Fan et al., 2019).The motivation of the provided solutions (Dataset and Model) was that the existing systems focus only on the sentence level while doing the extraction. This leads to a loss of relevant information that exists in the surrounding context.[The authors of this paper have presented two problems that are related to the sentence-level extraction, which are : Part-of-speech ambiguity and syntactic ambiguity.The part-of-speech ambiguity is related to the function of the word in the sentence. The syntactic ambiguity does manifest when we can not find an explicit relationship between two words]sentence-levelProblems.In both cases, looking at the surrounding context may resolve the problem.[For their new presented task]newTask[, a document-level annotated dataset was required.But all the existing datasets consider only the sentence level and does not look at the surrounding context.Therefore, a **Doc**ument-level context-aware **O**pen **I**nformation **E**xtraction (**DocOIE**) was presented.In fact, DocOIE is a dataset composed of 800 expert-annotated sentences from both healthcare and transportation fields]DocOIEIntro.[In addition to that, they have developed a **Doc**ument-level context-aware **O**pen **I**nformation **E**xtraction(**DocIE**)model]DocIEInto.The results show that it is crucial to consider the context and how this leads to more accurate tuple extraction. Related works will be discussed in the next Section.The methodology used during the dataset creation and the model development will be explored in Section 3.Followed by the obtained results in Section 4. Lastly, a critical analysis will be demonstrated in Section 5 along with its relation to social good and possible future work.

2 Background

OpenIE Datasets: The earlier systems that came since the introduction of the Open Information Task by (Yates et al., 2007) used small datasets without any evaluation of the performance.The first large Dataset was OIE2016 (Stanovsky and Dagan, 2016) which came with 3200 automatic generated sentences and a scoring framework. After that came Wire57(Lechelle et al., 2019) and has introduced 57 manually annotated sentences.A more recent dataset was CaRB(Bhardwaj et al., 2019) with 1282 annotated sentences with crowdsourcing and 50 expert-annotated sentences. [The number of the presented annotated sentences remain small and there is no referring to the contextual information during the annotation process]nbAnnotSent.

OpenIE Models: The first model that was introduced in the open Information Extraction area was Text Runner(Yates et al., 2007), followed by many other systems such us Reverb(**?**) and Openie4(Mausam, 2019a).Those systems are based on rules created by humans and on statistical methods. Furthermore, they require a prior syntactic and semantic analysis. More recent models are the neural OpenIE systems that have shown very good results.Those models do not require any prior analysis, but rather need training with numerous samples.However, it is unrealistic and really expensive to annotate manually about 100K of sentences.[So bootstrapping strategy was adopted in various works]bootstrappingStrategy. For example,(Mausam, 2019b) used generated pseudo labels by Openie4 and (Kolluru et al., 2020) used labels from multiple OpenIE systems.At this point, we have to admit none of all these systems considers document-level context in the tuple extraction, they only focus on the sentence-level.

3 Methods

This section describes the approach applied in this paper. First, the datasets are described, then their annotation process along with the consistency measurement result are reported and finally the developed neural model is represented.

3.1 Methods of DocOIE Dataset

Regarding the dataset, the authors of the paper have presented the data selection and the annotation process realized by the annotators.In addition to that, they have evaluated their work and proved the high-level annotation consistency. [The **D**ocument-level context-aware **O**pen **I**nformation **E**xtraction (**DocOIE**) consists mainly of two datasets]DocOIEDatasets :

The evaluation dataset [contains 800 expert-annotated sentences]evaluationNbSentences from 80 documents from both healthcare and transportation fields.

The training dataset is composed of, [2400 documents from both domains]trainingDoc.

3.1.1 Dataset Collection

[In this paper, the types of documents were specified and restricted to documents such as news and Wikipedia articles]docTypes.[They have decided to collect patent documents from PatFT [1]]docSource. [This choice was based mainly on 4 criteria :

Adequacy : that means reasonable number of sentences to provide sufficient context

Consistency : The sentence's correlation is important while deriving the proper context

Informativeness : Informative entities like relations and events are more useful for the extraction process

Syntactic Variety : the structure variety of the sentences help to improve the performance of the model by providing different scenarios]docSelectionCriteria.While using the PatFT search engine, keywords were required to retrieve the documents.For that, they have considered two criteria :

Magnitude : the keywords should provide sufficient number of patent documents

Diversity : the collected documents should be related to diversified inventions, organizations, dates, etc.

As a result, they have decided to use 'healthcare', 'traffic' and 'transportation' keywords keysCriteria. To guaranty the consistency of data, a cleaning process was elaborated.The first step was removing non-textual components.The second step was removing the 10% shortest and 10% longest documents.The remaining documents will be divided to documents for annotation (for the model evaluation) and training documents.

3.1.2 DocOIE Evaluation Dataset Selection

The annotation of the evaluation dataset is done with context-Awareness.In fact, the annotator needs to read the surrounded sentences or even the entire document.[To be able to recover a sufficient number of documents, the authors have decided to choose randomly 10 sentences from 80 documents for annotation]evalSentences.

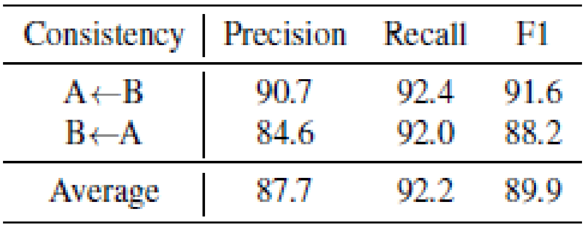
They randomly select 80 documents (40 in each field) and then randomly select 10 sentences from

[1] http://patft.uspto.gov/netahtml/PTO/search-bool.html

each document.As a result, we ended with a DocOIE evaluation dataset composed of 800 expert-annotated sentences.

3.1.3 Annotation Consistency Measurement

[The annotation process consists of three stages. As a first step, two annotators do annotations independently on 100 sentences among 800 sentences. Then they cross-validate and update the results. Secondly, the two annotators practice annotation independently on another 100 sentences among the remaining 700 sentences.After that, a consistency measurement is proceeded.This process is done to evaluate annotation agreement.For that, they have used an evaluation scorer proposed by CaRB(Bhardwaj et al., 2019).In this case, the scorer performs at the tuple level instead of the lexical level. In fact, they give a score for an expert's annotations by considering the other's annotations as gold annotations.Like Table 1 shows, this consistency measurement proves a high-level agreement between the two annotators, with an average F1 score of 89.9%. Based on this high-level consistency, the annotators annotate finally independently 300 sentences and then cross-validate and update the results]consistencyMeasurmentSteps.



Consistency	Precision	Recall	F1
A←B	90.7	92.4	91.6
B←A	84.6	92.0	88.2
Average	87.7	92.2	89.9

Table 1: Annotation consistency estimated between annotators A and B. A B indicates evaluation of A's annotations with B's annotations as ground truth.

3.1.4 Analysis of DocOIE Evaluation Dataset

[An analysis of the annotated sentences and tuples was elaborated in this paper. This helps in the understanding of the difficulty of DocOIE]analysisEvalDataset. The Analysis is performed in two levels :

Sentence-level Analysis : [evaluation of the complexity of the sentence based on the number of conjunction words, terminology mention and dependent clause]sentenceLevelAnalysis.

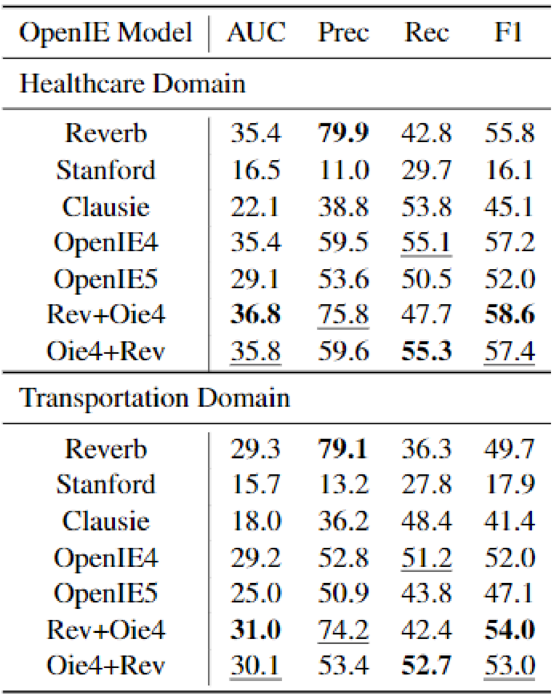
Tuple-level Analysis : [analysis of tuples based on possibility, negative polarity and underspecificity]tupleLevelAnalysis.

3.1.5 DocOIE Training Dataset

[For the training Dataset they have randomly selected 2400 documents (1200 each domain) from the documents collected from the patent PatFT]trainingDataset. The 1200 documents in each domain contain sufficient sentences (around, 120000 sentences) for the pseudo label generation required by openIE models.

3.2 Pseudo Label by Bootstrapping

In this paper, a neural openIE model will be trained, therefore many training labels are required. [The authors have decided to use the common practice (Kolluru et al., 2020) and CaRB scorer (Bhardwaj et al., 2019) to generate pseudo labels that will be considered as a training dataset]pseudoLabelStrategy. To guarantee a better quality of labels generation, they have proceeded an evaluation step of different traditional openIE models. The candidates were Reverb (Fader et al., 2011) , Clausie (Corro and Gemulla, 2013),Stanford OpenIE (Angeli et al., 2015),Ope-



OpenIE Model	AUC	Prec	Rec	F1
Healthcare Domain				
Reverb	35.4	**79.9**	42.8	55.8
Stanford	16.5	11.0	29.7	16.1
Clausie	22.1	38.8	53.8	45.1
OpenIE4	35.4	59.5	55.1	57.2
OpenIE5	29.1	53.6	50.5	52.0
Rev+Oie4	**36.8**	75.8	47.7	**58.6**
Oie4+Rev	35.8	59.6	**55.3**	57.4
Transportation Domain				
Reverb	29.3	**79.1**	36.3	49.7
Stanford	15.7	13.2	27.8	17.9
Clausie	18.0	36.2	48.4	41.4
OpenIE4	29.2	52.8	51.2	52.0
OpenIE5	25.0	50.9	43.8	47.1
Rev+Oie4	**31.0**	74.2	42.4	**54.0**
Oie4+Rev	30.1	53.4	**52.7**	53.0

Table 2: A Performance of OpenIE models on DocOIE evaluation dataset. The best scores are in boldface, and second-best scores are underlined.

nIE4 (Mausam, 2019a) and OpenIE5[2].

In addition to those five models, they have also evaluated the combinations of Reverb and OpenIE4.So in the case of Reverb+OpenIE4, reverb is the main system and if it fails then the extraction is completed with OpenIE4. And OpenIE4+Reverb uses OpenIE4 as the main system.The results are reported in the Table 2 and show that openIE4 and Reverb have the best performance, and that their combinations lead to the second-best performance in both domains.

3.3 DocIE Model

After preparing the Datasets for the training and the evaluation steps, they have presented their **D**ocument level context-aware **O**pen **I**nformation **E**xtraction model, named **DocIE**.[The proposed model is composed mainly of two parts: source-context encoder and encoder-decoder]modelParts.

[As preparation, they have denoted their document as D={s_1,s_2,..., s_N}(N is the number of sentences).They consider each s_i as the input where the tuples will be extracted from.But for each sentence, s_i they also look into its surrounding context c_i= {s_{i-t}, ...,s_{i-1},s_i,s_{i+1},...,s_{i+t}} where t represents the context window size]document-levelContext.

Source-context encoder Their source context encoder is basically inspired by a recent work (Ma et al., 2020) which was developed for machine translation. In this work, a flat-Transformer was adopted to incorporate the context in the source sentence.Based on this idea, DocIE encoder is composed of bottom and top blocks.[The bottom blocks take the concatenation of the context with the source sentence as input.The top blocks take only the new representation of the source sentence from the bottom blocks.To perform the semantic interaction between the context and the source sentence, BERT (Jacob Devlin, 2019) was used in the implementation]source-contextEncoder.

First, they project s and c into embedding space by making the sum of their word and segment embedding.Formally, we will get e_s=E(s)+S(s) and e_c=E(c)+S(s).E is the trainable word embedding matrix and S is the trainable segment embedding matrix. S is used to make the difference between the words from the source sentence and the words from the context sentence. In more details, the words from the source sentence are initialized to 0 and the ones from the context are initialized to 1.The concatenation of e_s and e_c [e_s,e_c] is given as input for the source-context encoder. BERT merges the information that comes from both source and context sentences.For a better representation, they used the last hidden state h1[s; c] of BERT as the representation of the two concatenated input sequences.

$$h_1[s;c] = BERT([e_s;e_c]) \qquad (1)$$

A transformer (Vaswani et al., 2017) is added as top blocks on the top of BERT blocks in order to prepare the source sentence representation for the following encoder-decoder. They truncate the context sentence representation h1[c] from h1[s;c] to get the source sentence representation.(only h1[s] is kept)

$$h_2[s] = Transformer(h_1[s]) \qquad (2)$$

Encoder-Decoder [A copyAttention (Mausam, 2019b) mechanism was adapted by the encoder-decoder.This mechanism considers the OpenIE task as sequence-sequence generation task with copying mechanism.The encoder-decoder represents a variable length sequence as input, and this will be used in the decoder to generate the output. In order to align the encoder hidden state with the decoder hidden state, an attention mechanism (Bahdanau et al., 2015) was integrated in the framework.Since the tuple arguments are parts of the input, a copying mechanism (Gu et al., 2016) was applied.In fact, it helps by copying the output tuples directly from the input.encoderDecoderMechanism]

4 Results

This part is dedicated to the evaluation of the DocIE and its comparison with two baseline neural OpenIE models, which are CopyAttention+BERT and IMOJIE (Kolluru et al., 2020) .Since DocIE adopts the copyAttention mechanism (Mausam, 2019b) as its encoder-decoder module, we can consider BERT + CopyAttention mechanism as the base model, from which DocIE adds its context-Awareness.

4.1 Neural Baseline models

In order to choose the best pseudo labels generator, they have evaluated the baseline models while

[2] github:dair-iitd/openie-standalone

using different systems in the label-generation process.The evaluation is conducted on the DocOIE evaluation dataset with CaRB scorer. The experiments have shown (refer to table 3) that for both models, BERT+CopyAttention and IMOJIE, pseudo-labels generated by Rev+Oie4 lead to better results in the healthcare domain. However, in the transportation domain, Oie4+Rev shows a better performance.This difference is mainly related to the fact that sentences in transportation domain tend to contain slightly more conjunctions (e.g., multiple conjunctions in one sentence).Based on the results of the experiment, OpenIE4 system generally extracts more tuples than Reverb and provides higher recall.That's why, extractions in transportation field with more conjunctions may better match the tuples extracted by OpenIE4.Based on the delivered results, they have decided to use pseudo labels by Rev+Oie4 for healthcare domain, and pseudo labels by Oie4+Rev for transportation domain modelDecisionGeneration.

Neural OpenIE	Pseudo labels	Healthcare				Transportation			
		AUC	Prec	Rec	F1	AUC	Prec	Rec	F1
CopyAttention+BERT	OpenIE4	38.6	54.4	**51.6**	52.9	38.5	54.3	**57.6**	55.9
	Oie4+Rev	40.4	57.1	50.4	53.5	38.3	55.3	56.9	**56.1**
	Reverb	43.7	77.8	46.4	58.1	36.9	70.5	42.2	52.8
	Rev+Oie4	**46.8**	**77.9**	48.6	**59.8**	**40.3**	**72.1**	43.9	54.6
IMOJIE	OpenIE4	36.2	73.0	**47.7**	57.7	35.7	62.9	48.8	55.0
	Oie4+Rev	34.1	69.5	46.7	55.9	**35.8**	63.5	**49.2**	**55.5**
	Reverb	38.5	79.2	45.6	57.9	33.2	77.3	39.2	52.0
	Rev+Oie4	**39.7**	**80.1**	46.4	**58.7**	33.0	**77.4**	39.6	52.4

Table 3: Neural baseline models trained with different pseudo labels. The best scores of each model are in boldface.

4.2 DocIE Against Baselines

System	Healthcare				Transportation			
	AUC	Prec	Rec	F1	AUC	Prec	Rec	F1
Rev+Oie4	36.8	75.8	47.7	58.6	31.0	**74.2**	42.4	54.0
Oie4+Rev	35.8	59.6	**55.3**	57.4	30.1	53.4	52.7	53.0
CopyAttention+BERT	46.8	77.9	48.6	59.8	38.3	55.3	56.9	56.1
IMOJIE	39.7	**80.1**	46.4	58.7	35.8	63.5	49.2	55.5
DocIE w/o transformer	47.1	76.2	49.9	60.3	**38.5**	55.8	57.0	56.4
DocIE w transformer	**47.4**	74.4	51.3	**60.8**	**38.5**	56.0	**57.5**	**56.9**

Table 4: Results of DocIE and baselines. The best scores are in boldface, and second-best scores are underlined.

In order to evaluate the performance of the DocIE model, they have compared it with sentence-level OpenIE systems.[For that, they have considered two forms of DocIE model, with and without the top transformer]modelsToEvaluate.For the experiment process, they have considered a window size of 5 for the healthcare domain and a window size of 4 for the transportation domain.Based on the delivered results by the table 4, we can admit that the DocIE model with transformer have achieved a better performance in terms of AUC(47.4) and F1 (60.8) scores in both domains. In the second place, we find the DocIE without transformer, that achieves a score of AUC of 47.1 and F1 of 60.3.It has actually outperformed all sentence-level models.

4.3 Impact of Context Window Size

For a better understanding of the effect of the window size, they have made an experiment where they track the influence of the number of the contextual sentences on the model performance.In fact, they have considered window-sizes in the range of 1 to 6 and plot the F1 score for each corresponding size.The results (refer to figure 1) show that the optimal number of sentences to be considered is 5 for healthcare domain and 4 for transportation domain. [From this experiment, we can conclude that a window-size of 4-5 (8-10 context-sentences) is sufficient to reach optimal performance results. In fact, small window-sizes can lead to insufficient context-understanding and large window-sizes may lead to noise]windowSizeRes.

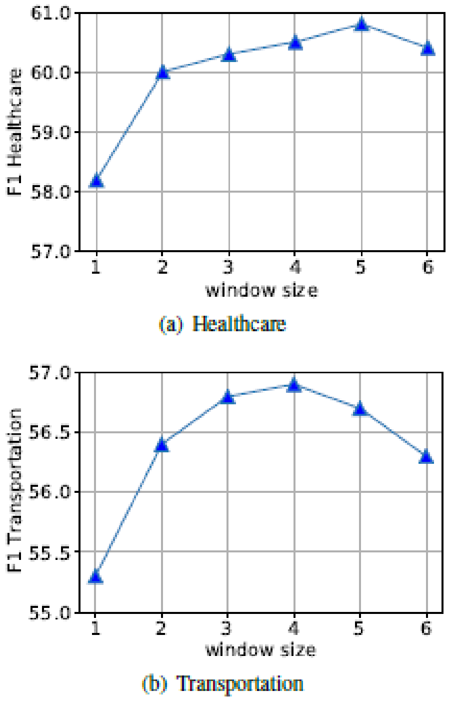


Figure 1: F1 with varying window sizes, on both domains.

4.4 Error Analysis

[In an error analysis similar to the one that was elaborated in (Kolluru et al., 2020) , they have evaluated the extracted tuples by DocIE from 50 randomly selected sentences in DocOIE. The major error types that were detected are]errorAnalysis :

Incompleteness: 28% of the sentences have represented an incomplete information extraction. In other words, DocIE does not extract correctly at least one key phrase in either arguments or relations.

Incorrect Boundary: 27% of the case made a misinterpretation of the syntactic meaning of the sentence

Redundant Extractions: 15% of the sentences, contain tuples that are extracted multiple times.

Grammatical Errors: 13% of extractions contain grammatical errors.

5 Analysis

In this section, the DocOIE dataset and DocIE model are critically assessed and the relation towards social good in the fields of positive outcomes and public policy is discussed.Possible future directions of research are also explored.

5.1 Shortcomings

In this paper, it was mentioned that the sentences were randomly selected from the documents. This may lead to an unbalanced complexity of the sentences.In other words, the selection may contain too long or too short sentences because no selection strategy was adopted. On the other hand, they have referred to a limited type of documents, which are the most informative types of documents like news and Wikipedia articles.Here, the performance of the model may be discussed because maybe we wouldn't get the same results if we considered for example social media information like comments.In fact, social media information tends to be noisy, informal, uncertain and does not have sufficient context information. Another point that has to be mentioned in this section is the limitation of the considered domains.Healthcare and transportation domains are the only two domains that were considered along this research. So to be able to make a generalization of the obtained results, more domains should be considered. Last but not least, pseudo-labels were generated using traditional models that do not have a context awareness. This may have strong influence on the DocIE model and limit its performance.

5.2 Relation to Positive Outcomes

Through a series of analysis and evaluation, the authors show the value and the potential to use the context while doing the extraction of the relational tuples. Unlike the other OpenIE models, using context is very efficient, and enables a more accurate tuple extraction. By using document-level information extraction, we can get more coherent and correct understanding about various topics, which helps us better address some social problems.For example, with the help of DocOIE and DocIE many scientific and historical articles could be correctly analyzed without a big risk of misinterpretation.It could also possibly help in improving the decision-making by the help of the correct information.

5.3 Relation to Public Policy

Many organizations are relying on NLP by extracting the relevant information.So extracting transparent and accurate information plays a huge role.Extracting correct information from sentences and reducing the error rate enables a better analysis and helps in writing the adequate public policy and taking the right measurement in many areas.

5.4 Future Work

In the future, we should consider more domains and more document types. In fact, like discussed in, (Habib and van Keulen, 2014) it will be more challenging to consider social media data, because this type of data is noisy and has short context.So it is really important to find a way to deal with them in order to guarantee the same performance as with the informative data. It is also interesting to investigate the possibility of not relying on the pseudo-labels used in the training phase because those are generated without any consideration of the context.

6 Conclusion

Even with some few shortcomings, the authors of this paper have proved that this new OpenIE task should be considered, and they have shown how important it is to consider the context while doing the extraction.They have presented DocOIE, the first document-level context-aware dataset composed of 800 expert-annotated sentences from 80 documents.While preparing the dataset, they have adopted one strategy for a careful document selection and another strategy to guarantee a high-level annotation consistency.This dataset was also adopted for the evaluation process of neural OpenIE models and proved the importance of incorporating the context to achieve a better performance.In comparison to all sentence-level OpenIE models, DocIE has achieved promising results and has outperformed all the other models.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Sangnie Bhardwaj, Samarth Aggarwal, and Mausam Mausam. 2019. Carb: A crowdsourced benchmark for open ie. page 6262–6267.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. pages 1535–1545.

Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. 2019. Using local knowledge graph construction to scale seq2seq models to multi-document inputs. page 4186–4196.

Kiril Gashteovski, Rainer Gemulla, Bhushan Kotnis, Sven Hertling, and Christian Meilicke. 2020. On aligning openie extractions with knowledge bases: A case study. pages 143–154.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning.

Mena B. Habib and Maurice van Keulen. 2014. Information extraction for social media.

Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

ATushar Khot, Ashish Sabharwal, and Peter Clar. 2017. Answering complex questions using open information extraction.

Keshav Kolluru, Samarth Aggarwal, Vipul Rathoreand Mausam, and Soumen Chakrabarti. 2020. Imojie: Iterative memory-based joint open information extraction.

William Lechelle, Fabrizio Gotti, and Phillippe Langlais. 2019. Wire57 : A fine-grained benchmark for open information extraction. pages 6–15.

Shuming Ma, Dongdong Zhang, and Ming Zhou. 2020. A simple and effective unified encoder for document-level machine translation.

Mausam. 2019a. Open information extraction systems and downstream applications.

Mausam. 2019b. Open information extraction systems and downstream applications.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. pages 2300–2305.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. 2007. Textrunner: Open information extraction on the web.